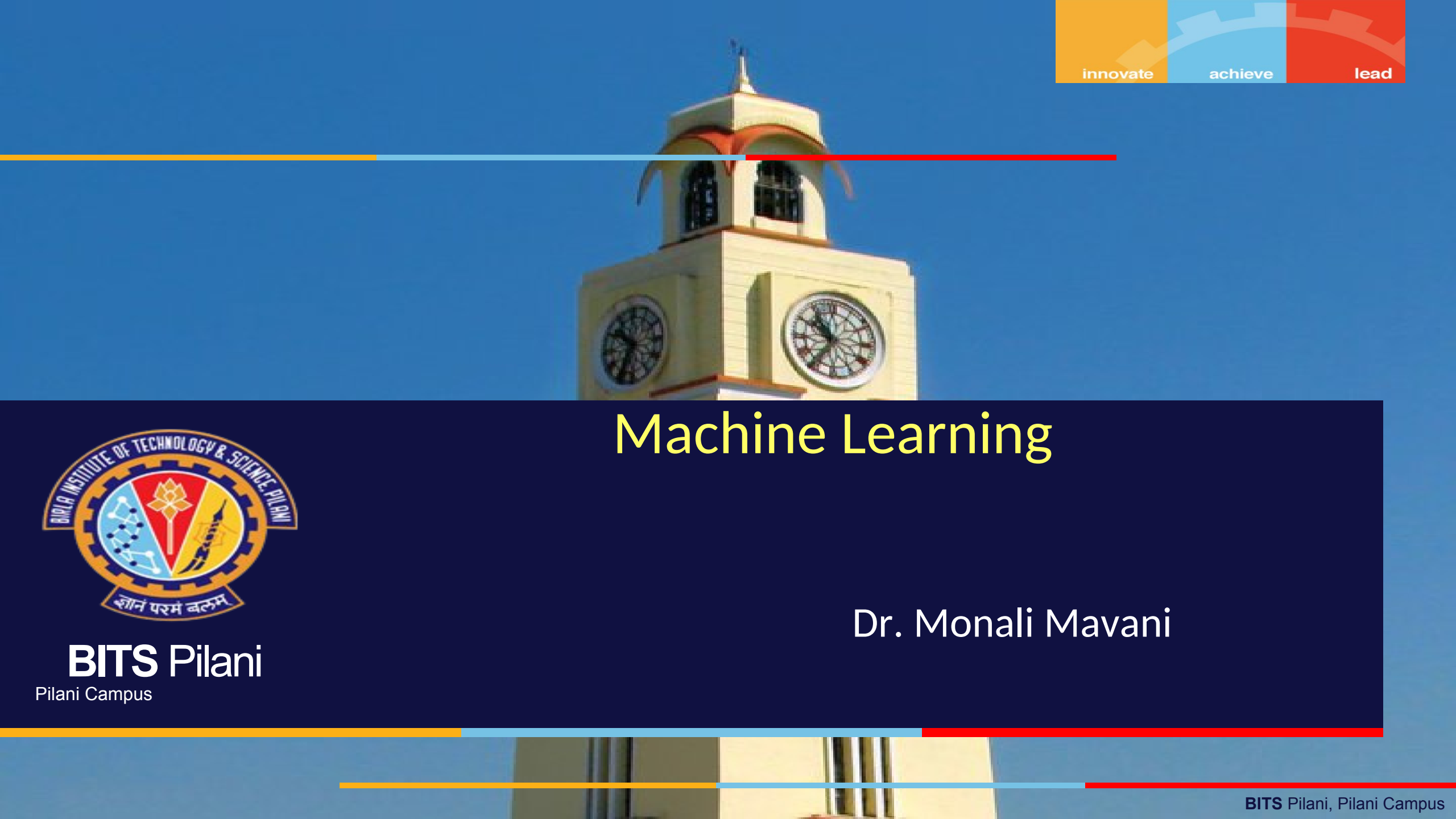


innovate achieve lead

Machine Learning

Dr. Monali Mavani

BITS Pilani
Pilani Campus

Disclaimer and Acknowledgement

- These content of modules & context under topics are planned by the course owner Dr. Sugata, with grateful acknowledgement to many others who made their course materials freely available online
- We here by acknowledge all the contributors for their material and inputs.
- We have provided source information wherever necessary
- Students are requested to refer to the textbook w.r.t detailed content of the presentation deck shared over canvas
- We have reduced the slides from canvas and modified the content flow to suit the requirements of the course and for ease of class presentation

Slide Source / Preparation / Review:

From BITS Pilani WILP: Prof.Sugata, Prof.Chetana, Prof.Rajavadhana, Prof.Monali, Prof.Sangeetha, Prof.Swarna, Prof.Pankaj

External: CS109 and CS229 Stanford lecture notes, Dr.Andrew NG and many others who made their course materials freely available online

Session Content

- Decision theory
  - Probabilistic generative models:
  - Probabilistic discriminative models
  - Discriminant functions
- Logistic regression
  - Cost function
  - Gradient descent : learning parameters
- Multi-class classification

Decision theory

- Probability theory provides a consistent framework for **quantifying and manipulating uncertainty**

- Decision theory combined with probability theory allows us to make **optimal decisions in situations involving uncertainty**

- We have an input vector x together with a corresponding target vector t
  - goal : predict t given a new value for x

- The joint probability distribution p(x, t) provides a complete summary of the uncertainty associated with these variables

Inductive Learning Hypothesis : Interpretation

- Target Concept : **t**
- Discrete : f(x) ∈ {Yes, No, Maybe} Classification
- Continuous : f(x) ∈ [20-100] Regression
- Probability Estimation : f(x) ∈ [0-1]

Sky	AirTemp	Humidity	Wind	Water	Forecast	*EnjoySport?*
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

Decision Theory

- Target Concept : t
- Discrete : f(x) ∈ {Yes, No} ie., t ∈ {0, 1}
- Continuous : f(x) ∈ [20-100]
- Probability Estimation : f(x) ∈ [0-1]

Binary Classification

ML Task : Predict the Employability of interview candidates based on CGPA & IQ

Preprocess Implemented:

Min-Max Normalization on IQ

CGPA	IQ	IQ	Job Offered
5.5	6.7	100	1
5	7	105	0
8	6	90	1
9	7	105	1
6	8	120	0
7.5	7.3	110	0

Decision Theory

$$p(X|c_k) = \frac{p(X,c_k)}{p(x)}$$

The decision problem: given x, predict t according to a probabilistic model p(x, t)

- Target Concept : **t**
- Discrete : f(x) ∈ {Yes, No} ie., **t** ∈ {0, 1} $\Leftrightarrow \{C_1, C_2\}$
- Continuous : f(x) ∈ [20-100]
- Probability Estimation : f(x) ∈ [0-1]

$p(x, C_k)$ is the (central!) inference problem

CGPA	IQ	IQ	Job Offered	P(Job = 1)
5.5	6.7	100	1	0.8
X =< 5 ,	> 7	105	0	0.4
8	6	90	1	0.75
9	7	105	1	0.95
6	8	120	0	0.35
7.5	7.3	110	0	0.4

$P(X, C_1)$

$= P(C_k|X)$

$P(X, C_2)$

$P(X, C_1)$

posterior Prob.

Bayes' rule

We are interested in the probabilities of the two classes, given the input features

Using Bayes' theorem

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})}$$

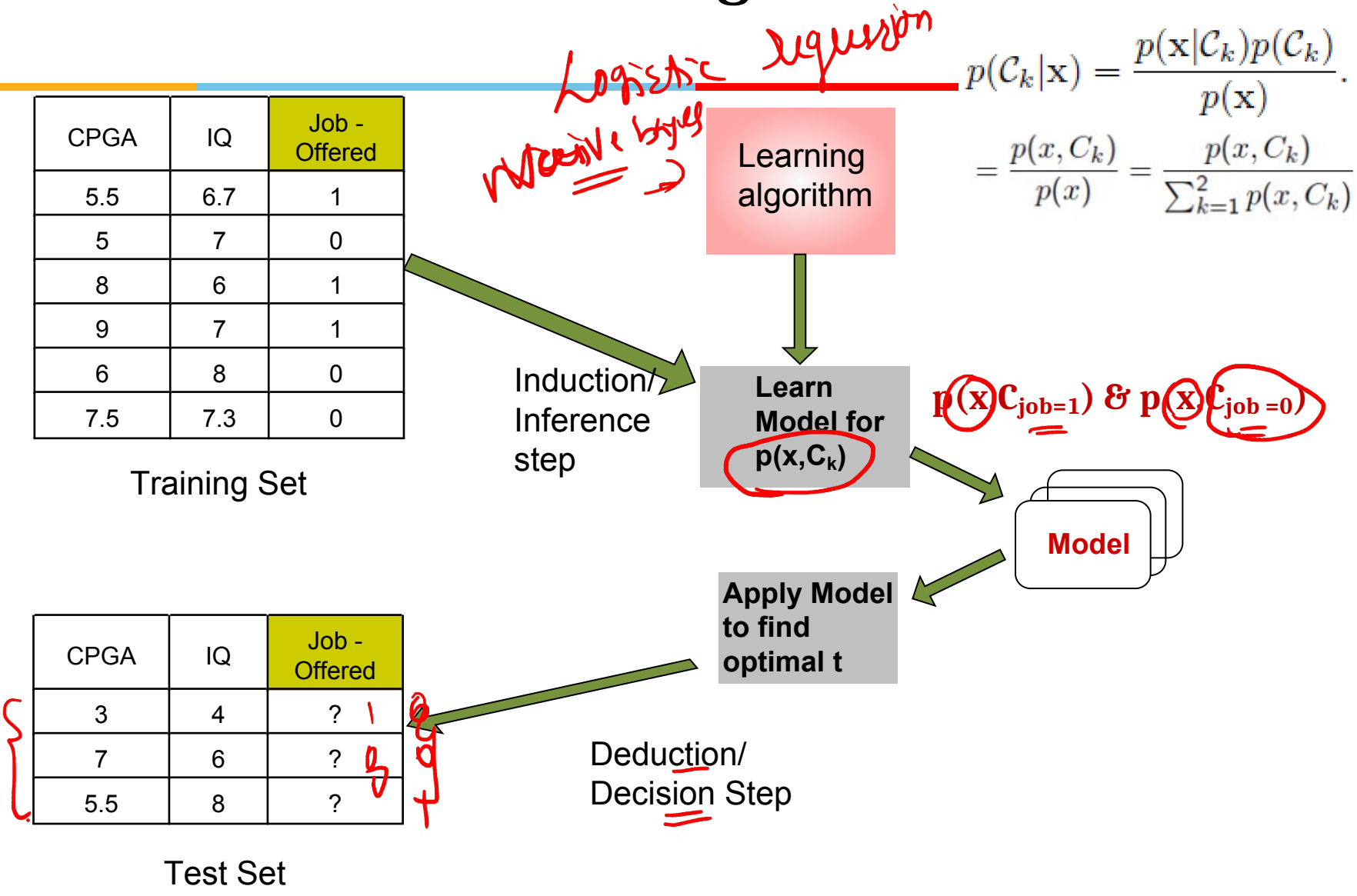
$p(C_k)$ = prior probability for class C_k and $p(C_k|x)$ = its corresponding posterior e.g.

- $p(C_1)$ = probability that the person enjoy sport, irrespective of days weather condition
- $p(C_1|x)$ = corresponding probability, revised using Bayes in the light of weather information

If our aim is to minimize the chance of assigning x to the wrong class, then intuitively we would choose the class having the highest posterior probability

This intuition is clearly correct, but why? – decision theory

Classification Problem: Stages



CPGA	IQ	Job - Offered
5.5	6.7	1
5	7	0
8	6	1
9	7	1
6	8	0
7.5	7.3	0

Training Set

CPGA	IQ	Job - Offered
3	4	?
7	6	?
5.5	8	?

Test Set

x => height

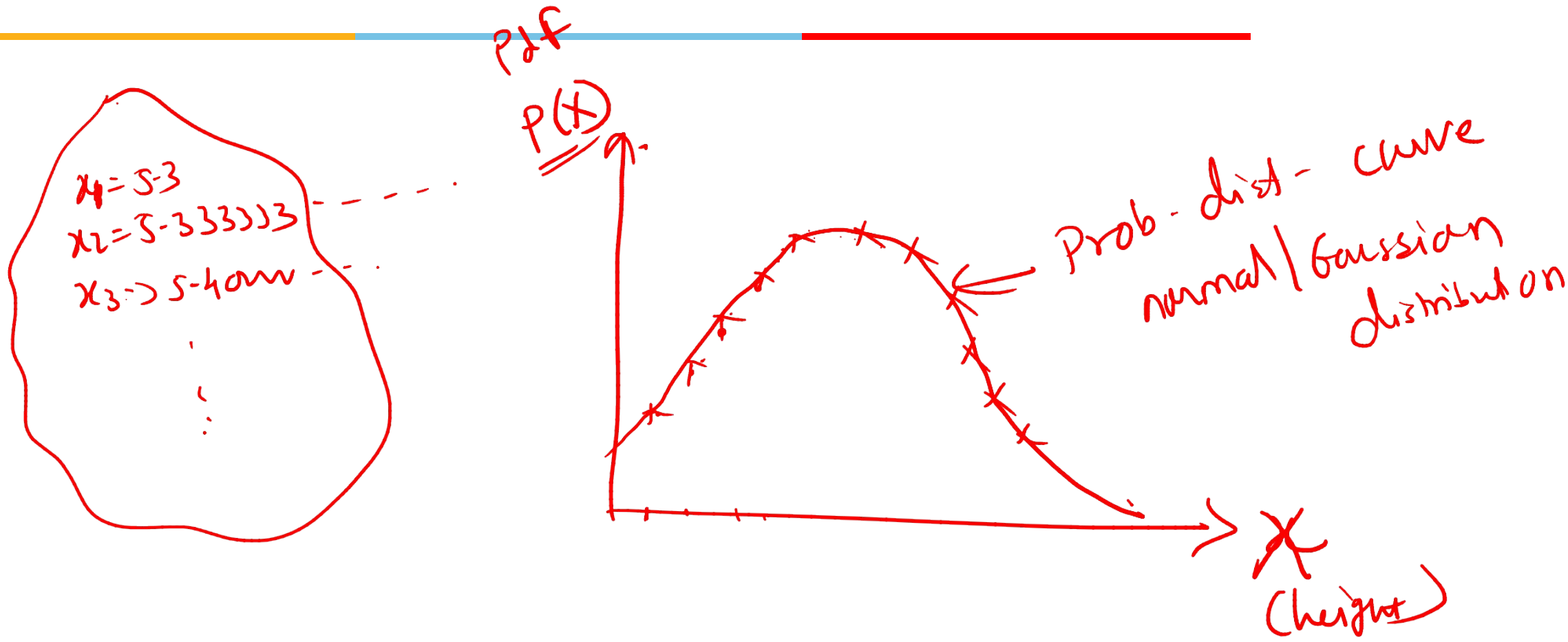

pdf
P(x)
x1 = 5-3
x2 = 5-333333
x3 -> 5-4ooo
Prob-dist-curve
normal/Gaussian
distribution
X
(height)

Illustration of the joint probabilities $p(x, C_k)$

Training Set

CPGA	IQ	Job - Offered
5.5	6.7	1
5	7	0
8	6	1
9	7	1
6	8	0
7.5	7.3	0

Induction/ Inference step

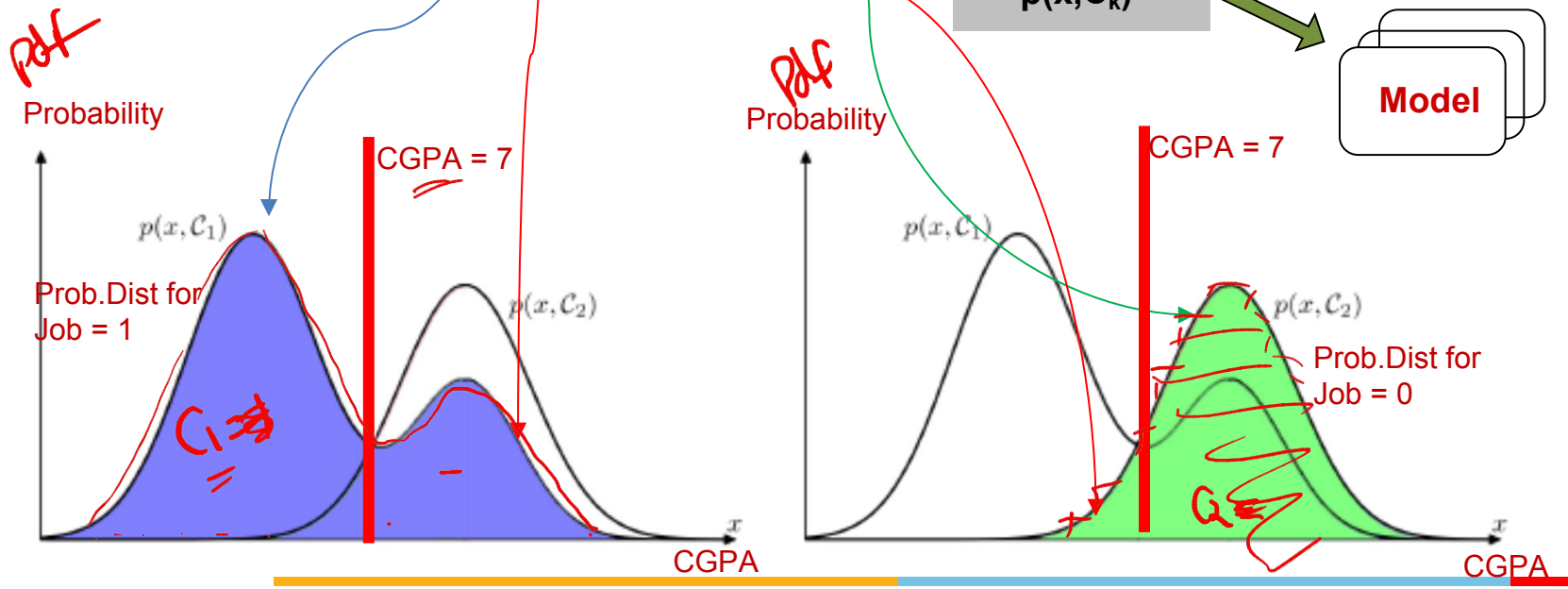
Learning algorithm

Learn Model for $p(x,C_k)$

Model

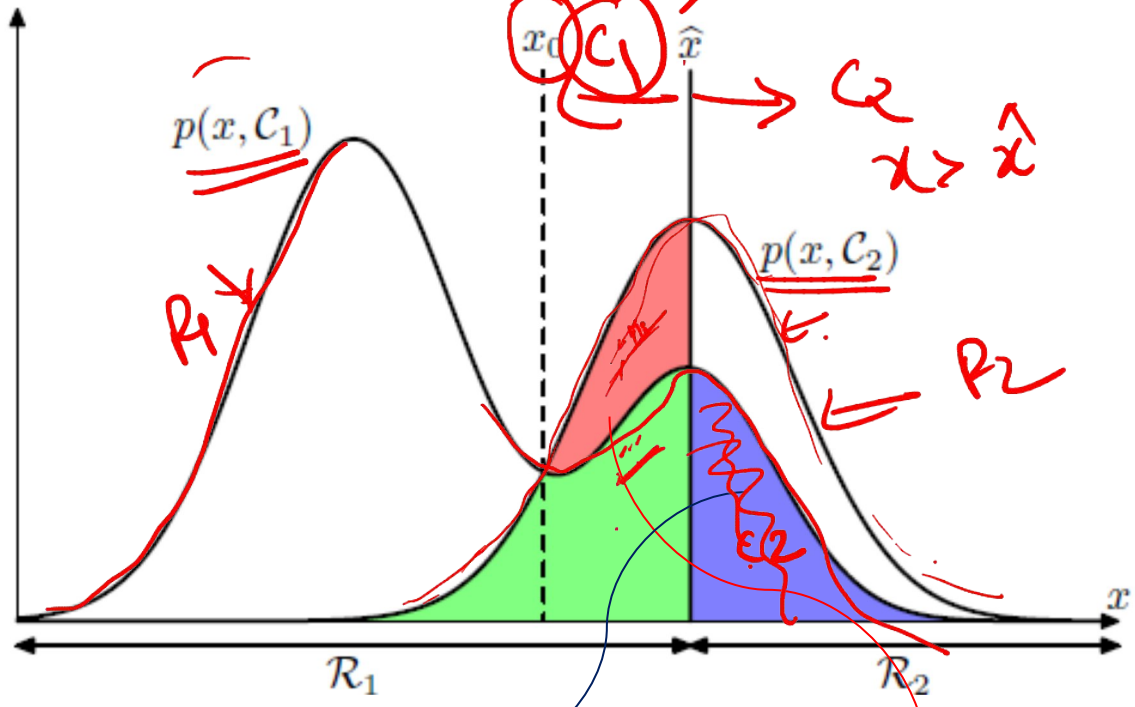
if CGPA ≥ 7 => C2

if CGPA < 7 => C1



Decision Regions

Sample Rule / Hypothesis:
IF CGPA>7 Job = 1
Else Job = 0

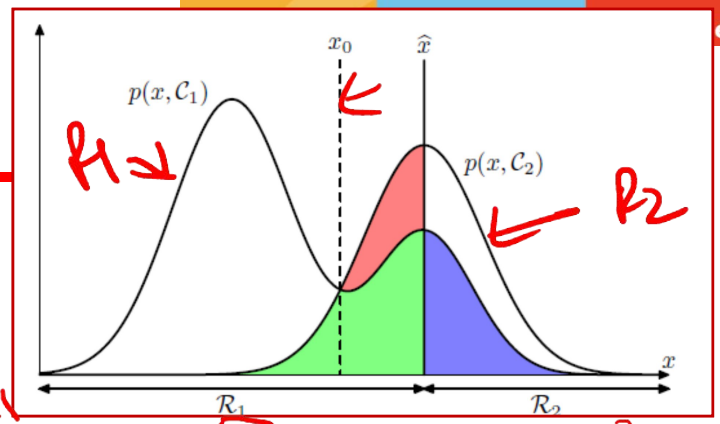


$$p(\text{mistake}) = p(x \in \mathcal{R}_1, \mathcal{C}_2) + p(x \in \mathcal{R}_2, \mathcal{C}_1)$$
$$= \int_{\mathcal{R}_1} p(x, \mathcal{C}_2)\,dx + \int_{\mathcal{R}_2} p(x, \mathcal{C}_1)\,dx.$$

- Goal: make as few misclassifications as possible
- Need a rule that assigns x to one of the available classes
- Such rule divides the input space into regions $\mathbf{R}_k$ called **decision regions**, one for each class, such that all points in R_k are assigned to class C_k
- The boundaries between regions are called decision boundaries
- For $x < \hat{x}$ errors represented by sum of the red and green regions (errors are due to points from class C_2 being misclassified as C_1)
- For $x > \hat{x}$ errors represented by the blue region (errors are due to points from class C_1 being misclassified as C_2)

Minimum Misclassification Rate

$$p(\text{mistake}) = p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1)$$
$$= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2)\, d\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1)\, d\mathbf{x}.$$



- Minimum error decision rule
  - For a given x choose class for which integrand is smaller
  - Since $p(x,C_k)=p(C_k|x)p(x)$,
  - choose class for which a posteriori probability is highest
  - Called Bayes Classifier

$$p(x, C_1) > p(x, C_2)$$
$$\Leftrightarrow p(C_1|x)p(x) > p(C_2|x)p(x)$$
$$\Leftrightarrow p(C_1|x) > p(C_2|x)$$

$$p(C_k|x) = \frac{p(x, C_k)}{p(x)}$$

Decision theory : Interpretation

We have broken the classification problem down into two separate stages

- Inference: we use training data to learn a model for $p(C_k|x)$
- Decision: we use $p(C_k|x)$ to make optimal class assignments

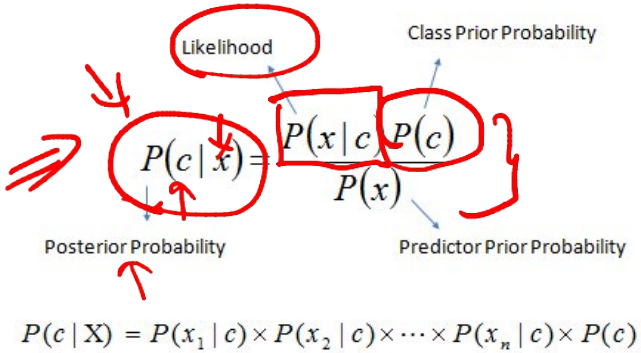
choose class for which a posteriori probability is highest

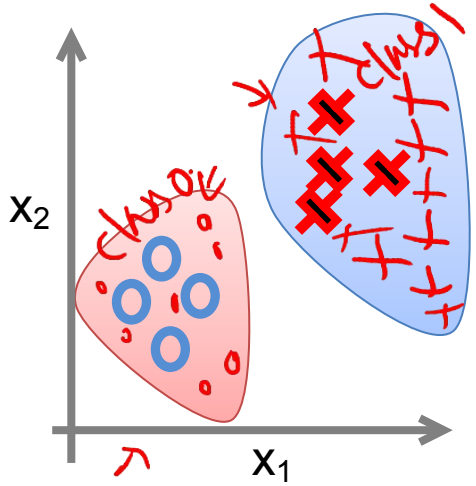
Types of classification models

1. Rely on a probabilistic model, with 2 flavours:
   - Generative:
   - Discriminative
2. Discriminant functions

Types of Classification - Generative

1. Solve the inference problem of determining the class-conditional densities $p(X|C_k)$ for each class C_k individually
2. Separately infer the prior class probabilities $p(C_k)$
3. Use Bayes' theorem to find posterior class probabilities $p(C_k|x)$

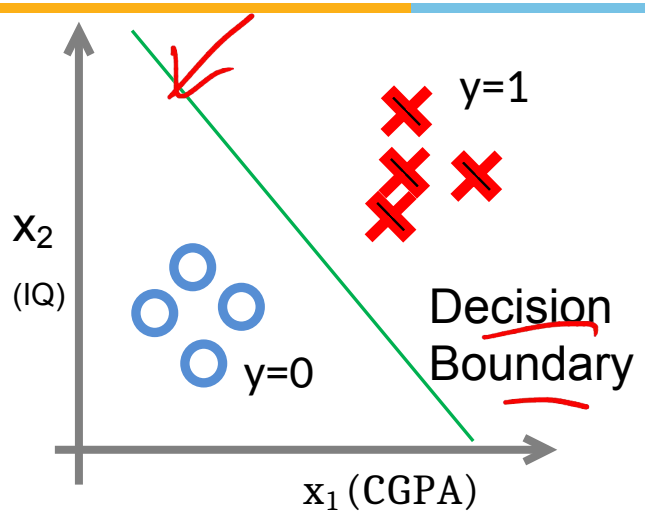




CPGA	IQ	Job - Offered
5.5	6.7	1
5	7	0
8	6	1
9	7	1
6	8	0
7.5	7.3	0
...	....	.....

- Known as generative models, because by sampling from them it is possible to generate synthetic data points in the input space.
- Eg., Gaussians, **Naïve Bayes**, Mixtures of multinomials , **Mixtures of Gaussians**, Bayesian networks
- Estimating the joint probability p(x, Ck) can be computational and data demanding
- High dimensionality, and consequently we may need a large training set

Types of Classification: Discriminative



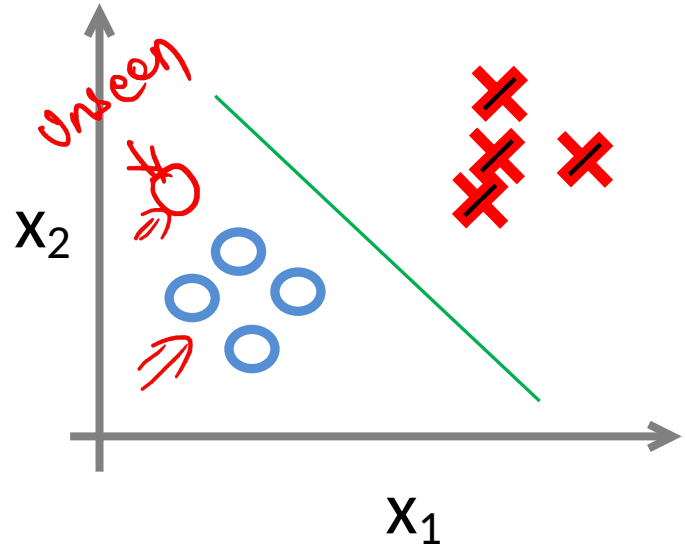
CPGA	IQ	Job - Offered
5.5	6.7	1
5	7	0
8	6	1
9	7	1
6	8	0
7.5	7.3	0
…	….	…..

- Infer directly $p(C_k|x)$
  - No attempt to model underlying probability distributions
  - we only really need the posterior probabilities
- Logistic regression, SVMs , tree based classifiers (e.g. decision tree), traditional neural networks, Nearest neighbor

Example: Spam classification problem

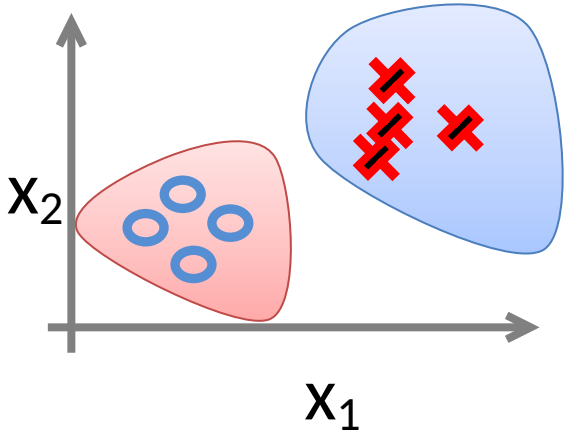
First Strategy: discriminative (e.g., logistic regression)

- Use training set to find a decision boundary in the feature space that separates spam and non-spam emails
- Given a test point, predict its label based on which side of the boundary it is on.

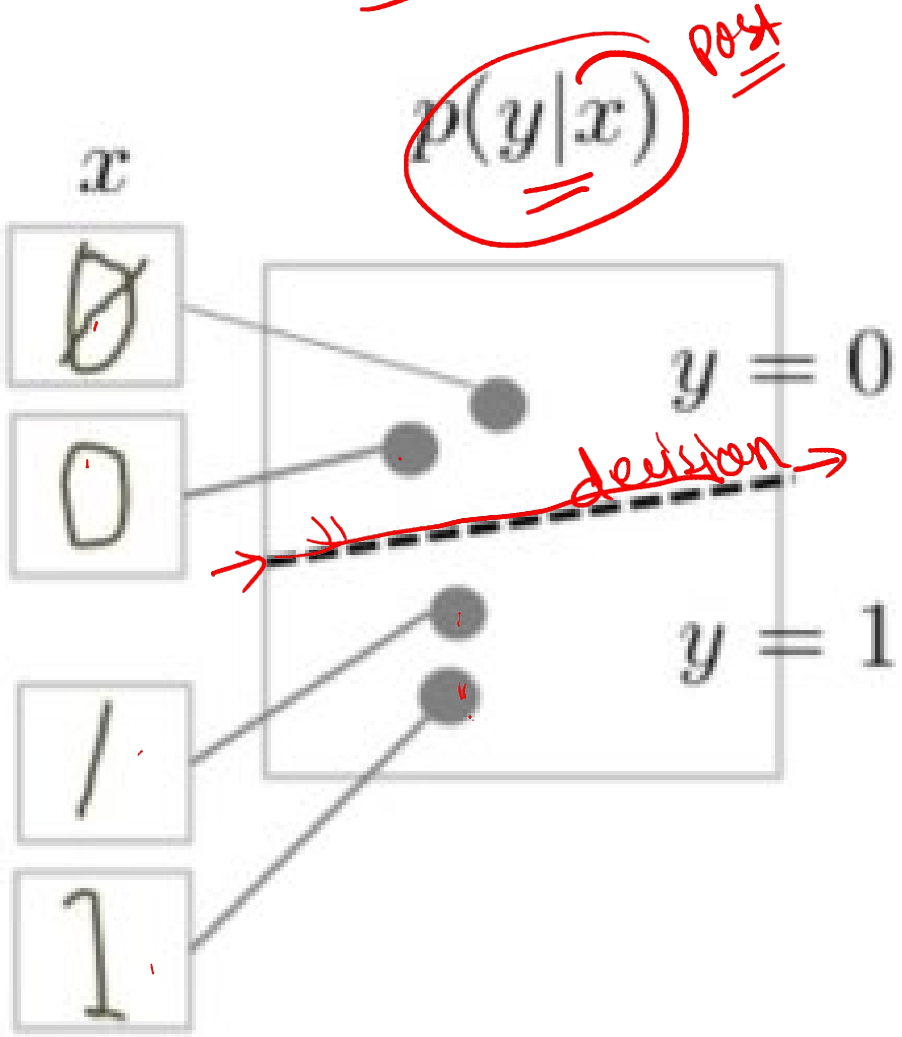


Second Strategy: generative (e.g., naive bayes)

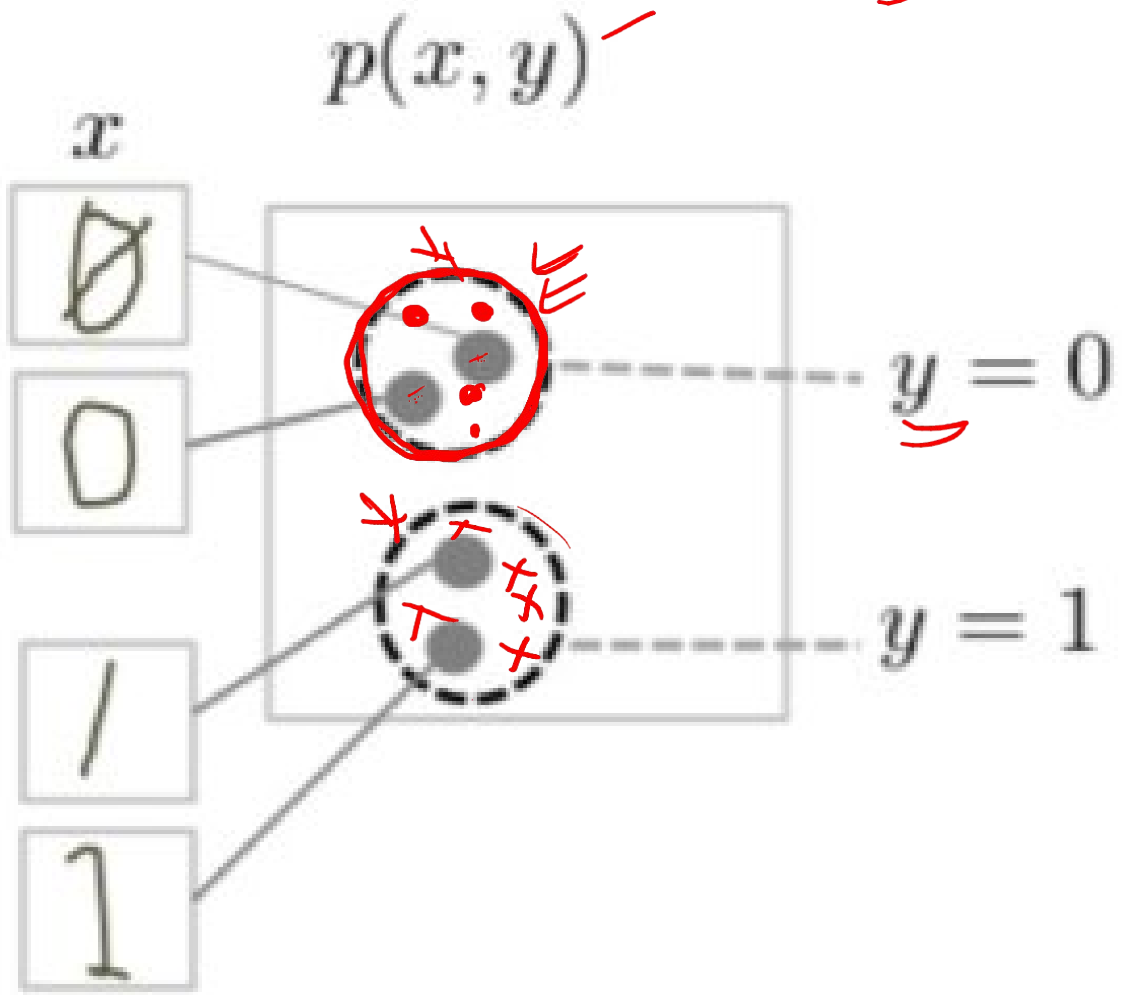
- Look at spam emails and build a model of what they look like.
- Similarly, build a model of what non-spam emails look like.
- To classify a new email, match it against both the spam and non-spam models to see which is the better fit.



- Discriminative Model



- Generative Model

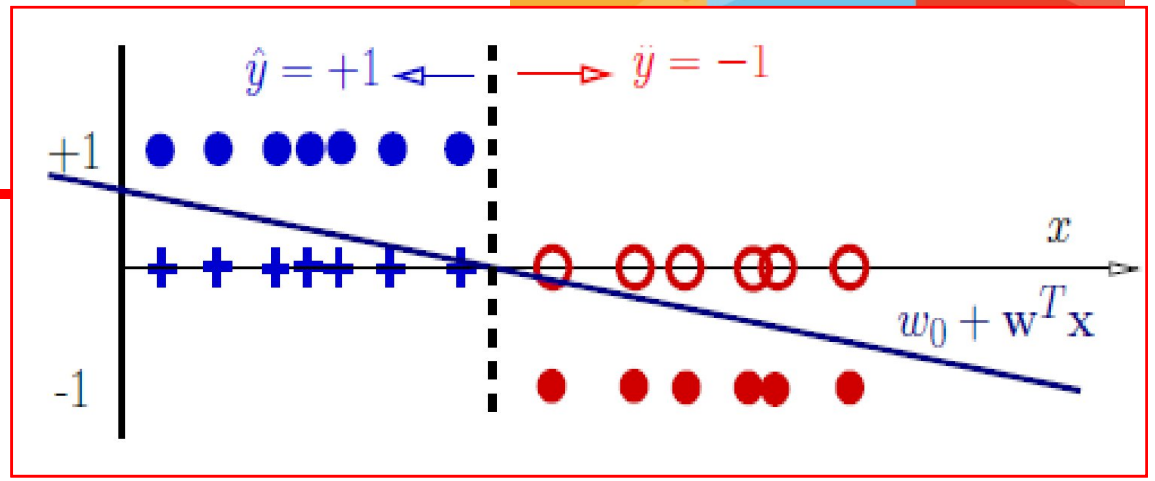


Types of Classification: Discriminant function

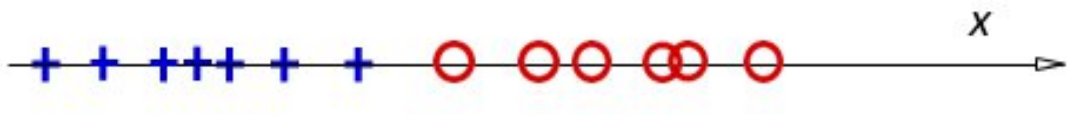
- There is an alternative, solve inference and decision together
  - We learn a function that maps inputs into decisions (discriminant function)
  - Inference and decision are combined into a single learning problem
  - Probabilities play no role, No access to posterior probabilities $p(C_k|X)$
  - Find a discriminant function f (x) which maps each input x directly onto a class C_k

- Linear discriminant function : $\boldsymbol{f(x,w) = w_0 + w^T x}$
  - Decision rule: Mathematically $\boldsymbol{y = sign(w_0 + w^T x)}$
  - Decision surface $y = \begin{cases} 1 & \text{if } f(\mathbf{x}, \mathbf{w}) \geq 0 \\ -1 & \text{otherwise} \end{cases}$ defined by the relation $y(\mathbf{x}) = 0$
  - linear function of input X
  - Parameter learning of linear discriminant functions: least squares, Fisher's linear discriminant, perceptron algorithm

Classification as regression



One dimensional example (input x is 1-dim)



Classifier:

$f(\mathbf{x}, \mathbf{w}) = w_o + \mathbf{w}^T\mathbf{x}$ (linear discriminant function)

Decision rule is

$$y = \begin{cases} 1 & \text{if } f(\mathbf{x}, \mathbf{w}) \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

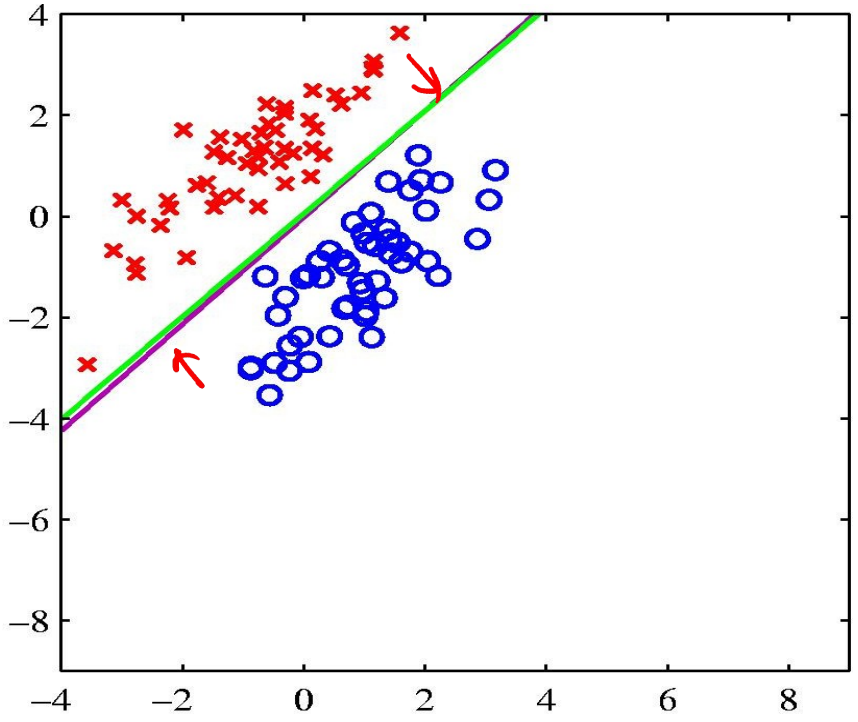
Mathematically

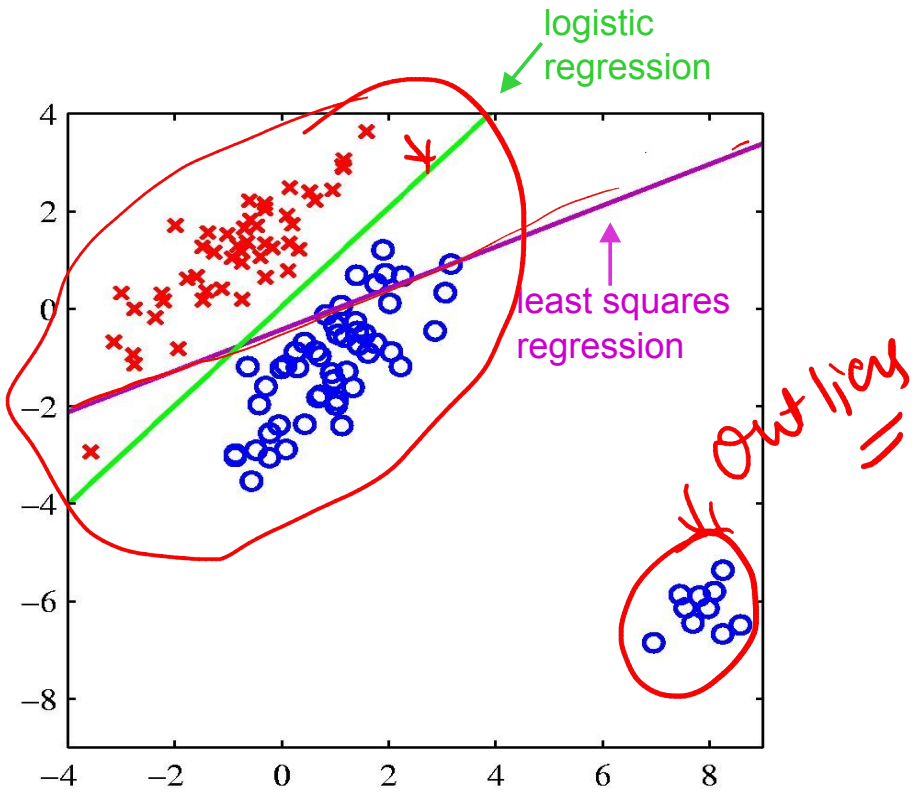
$$y = \text{sign}(w_0 + \mathbf{w}^T\mathbf{x})$$

This specifies a linear classifier: it has a linear boundary (hyperplane)

$$w_0 + \mathbf{w}^T\mathbf{x} = 0$$

Logistic Regression vs Least Squares Regression





The right-hand plot shows the corresponding results obtained when extra data points are added at the bottom left of the diagram, showing that least squares is highly sensitive to outliers, unlike logistic regression.

Logistic Regression

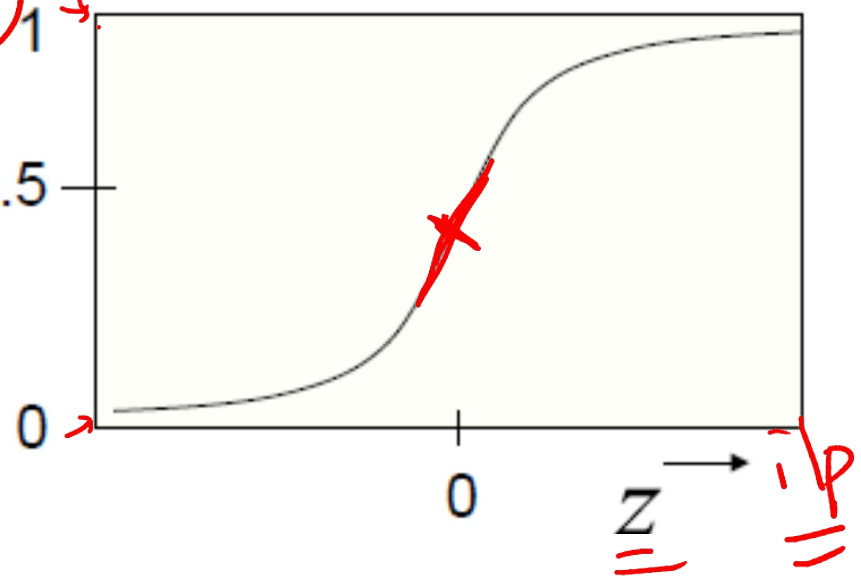
A probabilistic discriminative classifier, *generalized linear model*

Sigmoid applied to a linear function of the data

$$y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + w_0)$$

where the sigmoid:

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$



The output is a smooth function of the inputs and the weights

Features can be discrete or continuous

- Output of the model: value y ∈ [0, 1]
- Allows for **gradient-based learning of the parameters**

Sigmoid/Logistic Function

- Sigmoid/logistic function takes a real value as input and outputs another value between 0 and 1
- A useful property: easy to compute differential at any point
- That framework is called **logistic regression**
  - Logistic: A special mathematical sigmoid function it uses
  - Regression: Combines a weight vector with observations to create an answer

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

general symbol for any kind of non-linear function

$= \sigma(\theta^T x)$

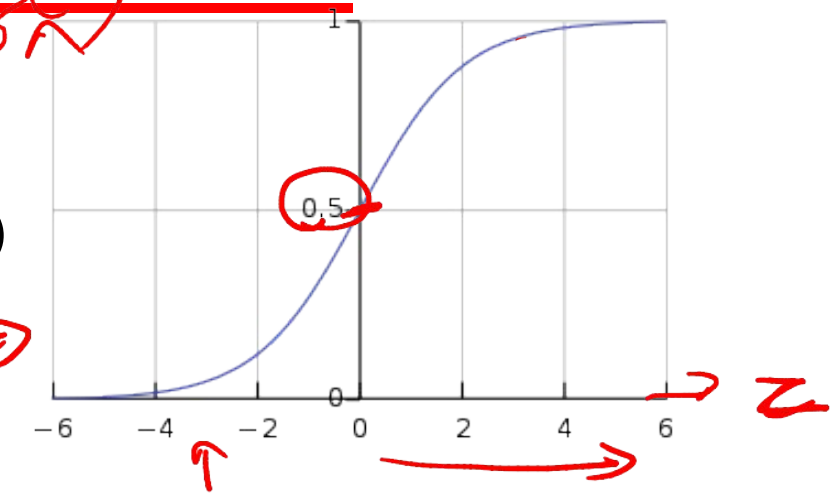
sigmoid symbol

Interpretation of hypothesis output

$$h_w(x) = g(w^\top x)$$

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

$$z = w^\top x$$



$$P(y = 1 \mid x; w) = h_w(x)$$

$$P(y=0 \mid x; w) = 1 - h_w(x)$$

(Recall decision surfaces correspond to *h*(x) = constant)

Suppose predict "y = 1" if $h_w(x) \geq 0.5$

$$z = w^\top x \geq 0$$

predict "y = 0" if $h_w(x) < 0.5$

$$z = w^\top x < 0$$

σ(z) is nonlinear, however, the decision boundary is determined by $w^T x = 0$, which is a linear function in x

Slide credit: Andrew Ng

Interpretation of hypothesis output

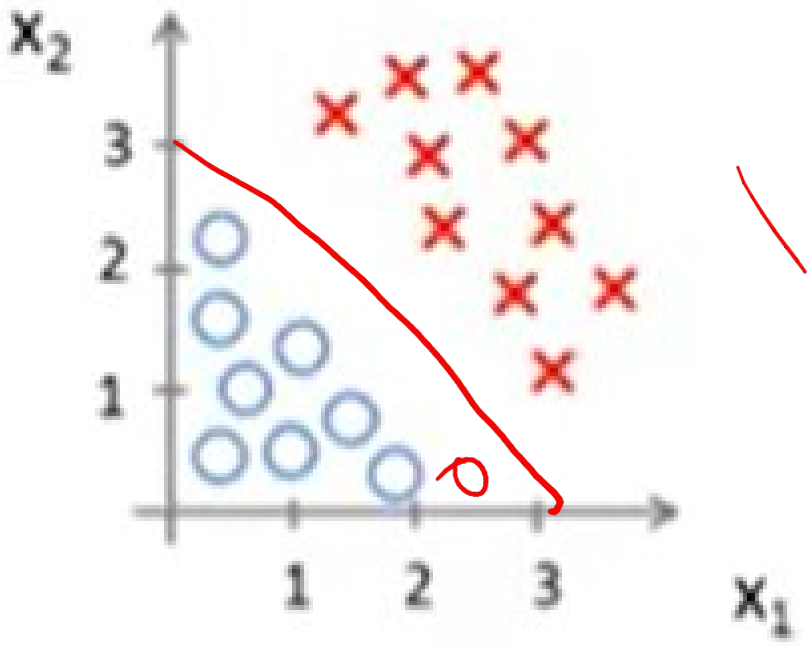
- $h_w(x)$ = estimated probability that $y = 1$ on input x
- Example: If $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$
- $h_w(x) = 0.7$
- Tell patient that 70% chance of tumor being malignant

$$h_w(x) = \frac{1}{1 + e^{w_0 + w_1 \cdot \text{tumorSize}}}$$

Slide credit: Andrew Ng

Decision boundary – logistic regression

$$z = w^T x \geq 0$$



$$h_w(x) = g(w_0 + w_1 x_1 + w_2 x_2)$$

E.g., $w_0 = -3,\ w_1 = 1,\ w_2 = 1$

Predict "$y = 1$" if $-3 + x_1 + x_2 \geq 0$

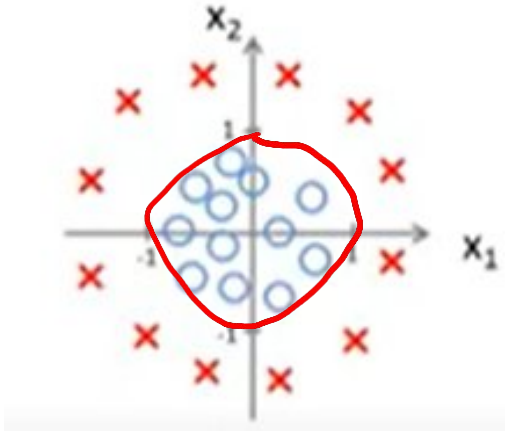
i.e $x_1 + x_2 \geq 3$

"$y = 0$" if $-3 + x_1 + x_2 < 0$

i.e $x_1 + x_2 < 3$

$x_1 + x_2 \geq 3$

Decision boundary – Fixed basis functions

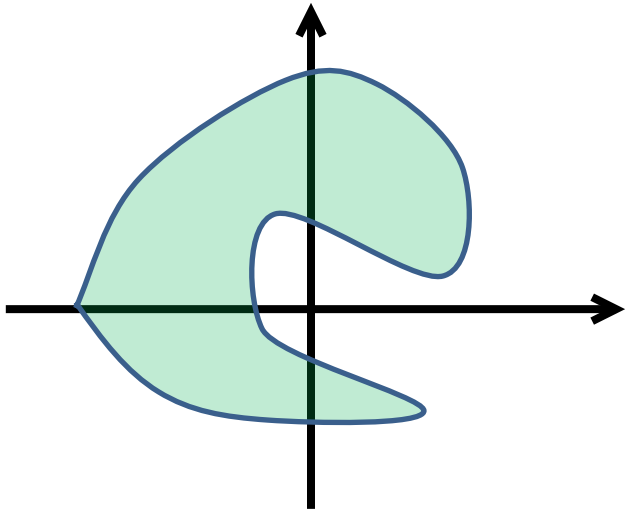


- $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$

E.g., $\theta_0 = -1,\ \theta_1 = 0, \theta_2 = 0, \theta_3 = 1, \theta_4 = 1$

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

$x_1^2 + x_2^2 \geq 1$



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

Slide credit: Andrew Ng

Learning model parameters

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

$$x \in \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix}$$

$x_0 = 1, y \in \{0, 1\}$

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

- How to choose parameters (feature weights) ?

MSE as a Cost Function

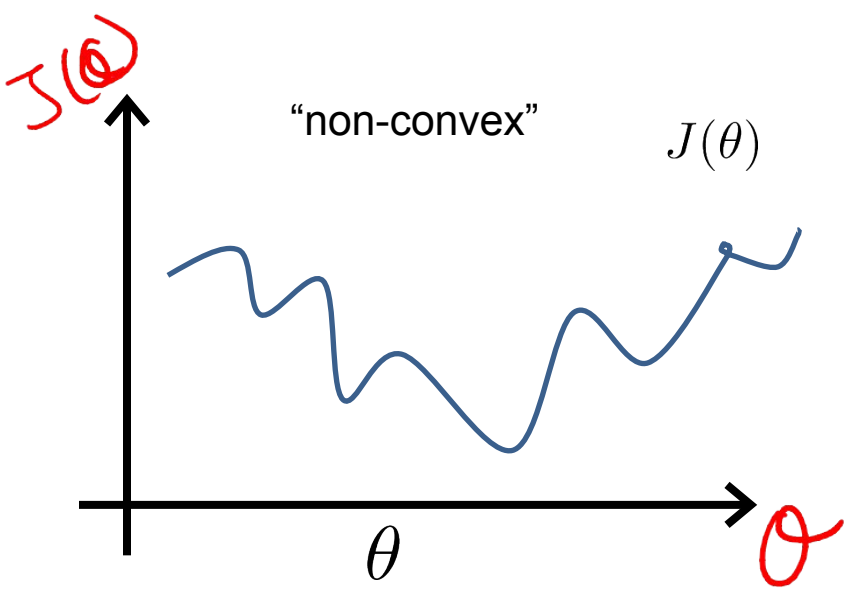
Linear reg:
$h_\theta(x) = \theta^T x$

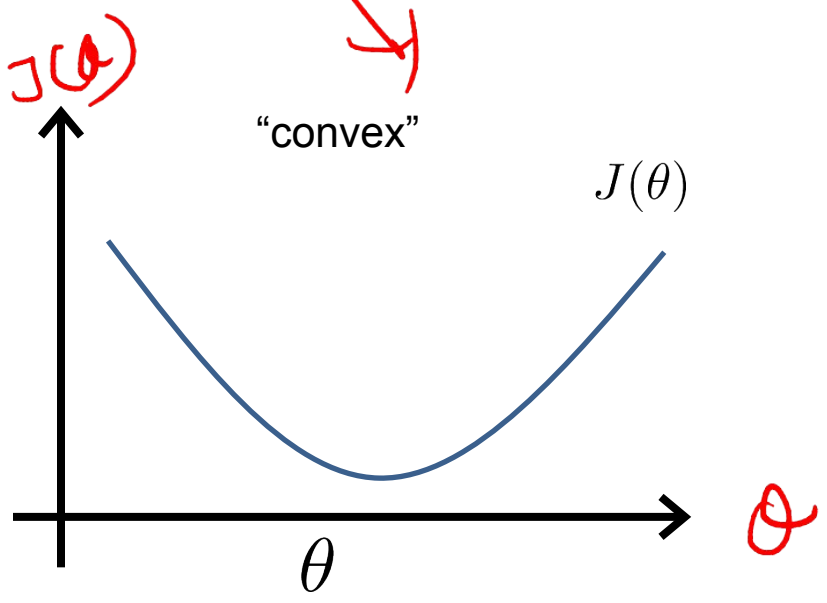
Logistic
$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

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

pred − act

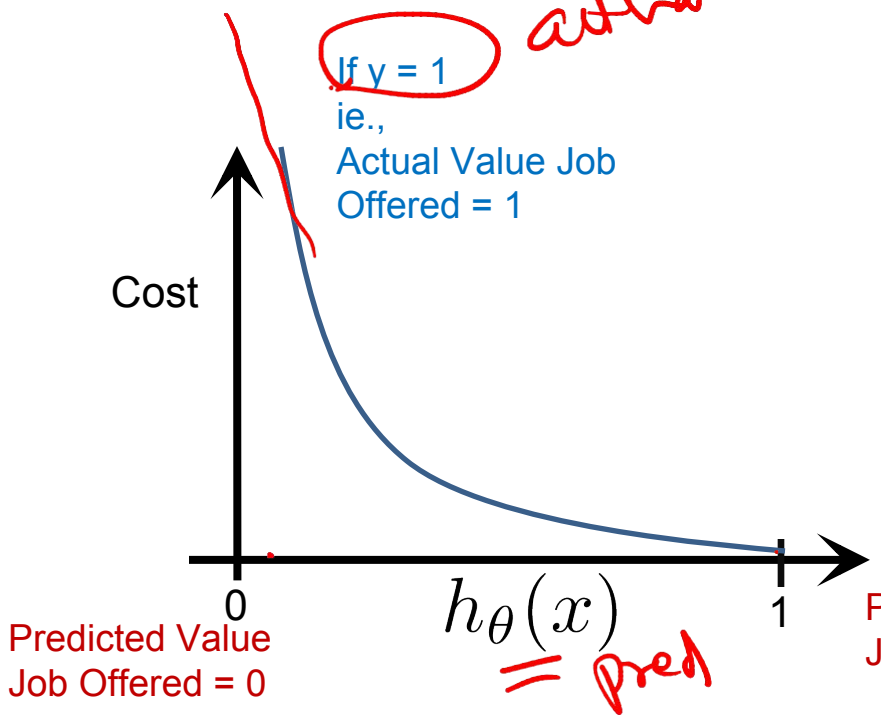
$$\text{Cost}(h_\theta(x^{(i)}), y^{(i)}) = \frac{1}{2} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$





Logistic regression cost function (cross entropy)

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

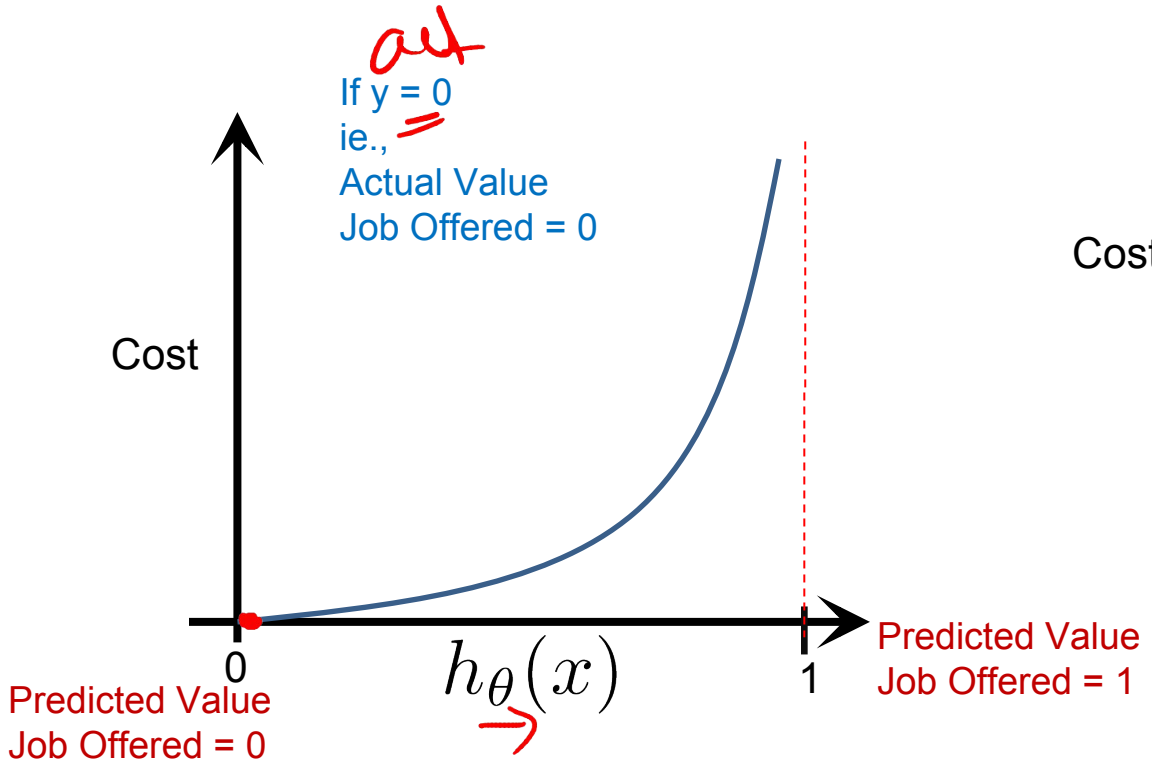


$\text{Cost} = 0$ if $y = 1, h_\theta(x) = 1$
But as $h_\theta(x) \to 0$
$Cost \to \infty$

Captures intuition that if $h_\theta(x) = 0$, (predict $P(y = 1|x; \theta) = 0$), but $y = 1$, we'll penalize learning algorithm by a very large cost.

Logistic regression cost function

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



Cost=0; If y=0 and $h_\theta(x)=0$

Cost function

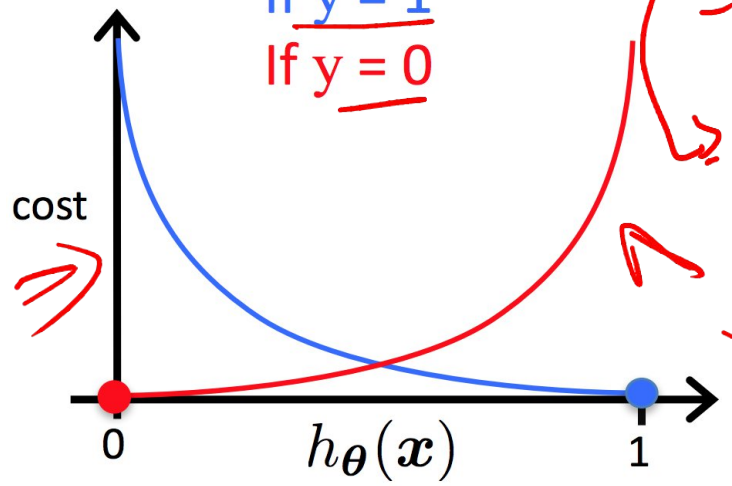
$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

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

To fit parameters θ : Apply Gradient Descent Algorithm $\min_\theta J(\theta)$

To make a prediction given new x :

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



Gradient descent

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

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

Repeat {

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

}

(Simultaneously update all θ_j)

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m}\sum_{i=1}^{m} (h_\theta\left(x^{(i)}\right) - y^{(i)}) x_j^{(i)}$$

Good news: Convex function!

Bad news: No analytical solution

Gradient descent for Linear Regression

Repeat {

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

}

$$h_\theta(x) = \theta^\top x$$

Gradient descent for Logistic Regression

Repeat {

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

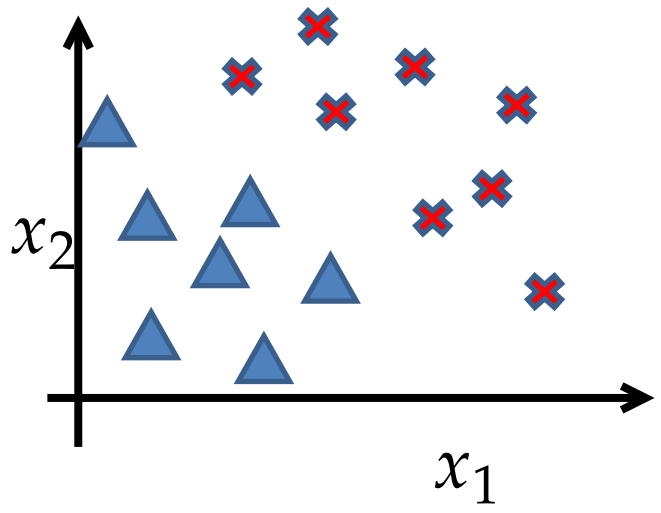
}

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

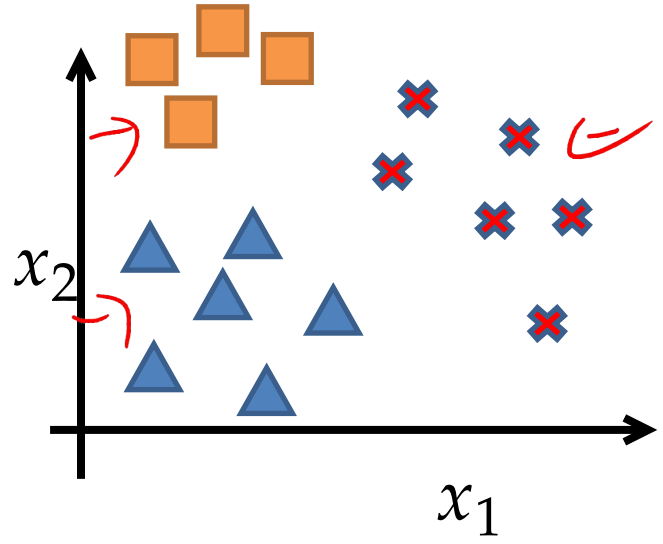
Multi-class classification

- Email foldering/tagging: Work, Friends, Family, Hobby
- Medical diagrams: Not ill, Cold, Flu
- Weather: Sunny, Cloudy, Rain, Snow

Binary classification



Multiclass classification



Slide credit: Andrew Ng

One-vs-all

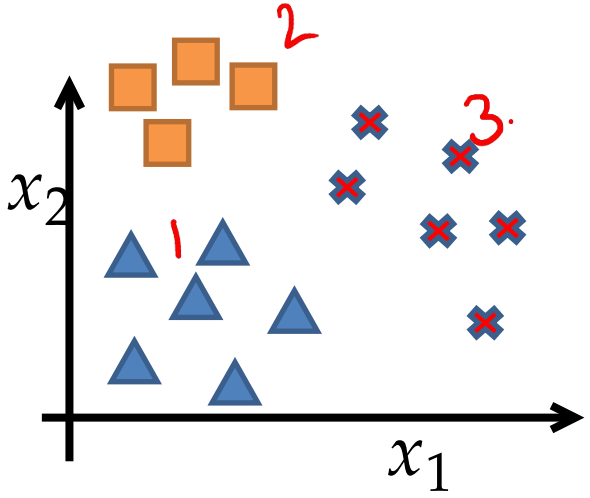
- Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y = i$

- Given a new input x, pick the class i that maximizes

$$\max_{i} h_\theta^{(i)}(x)$$

Slide credit: Andrew Ng

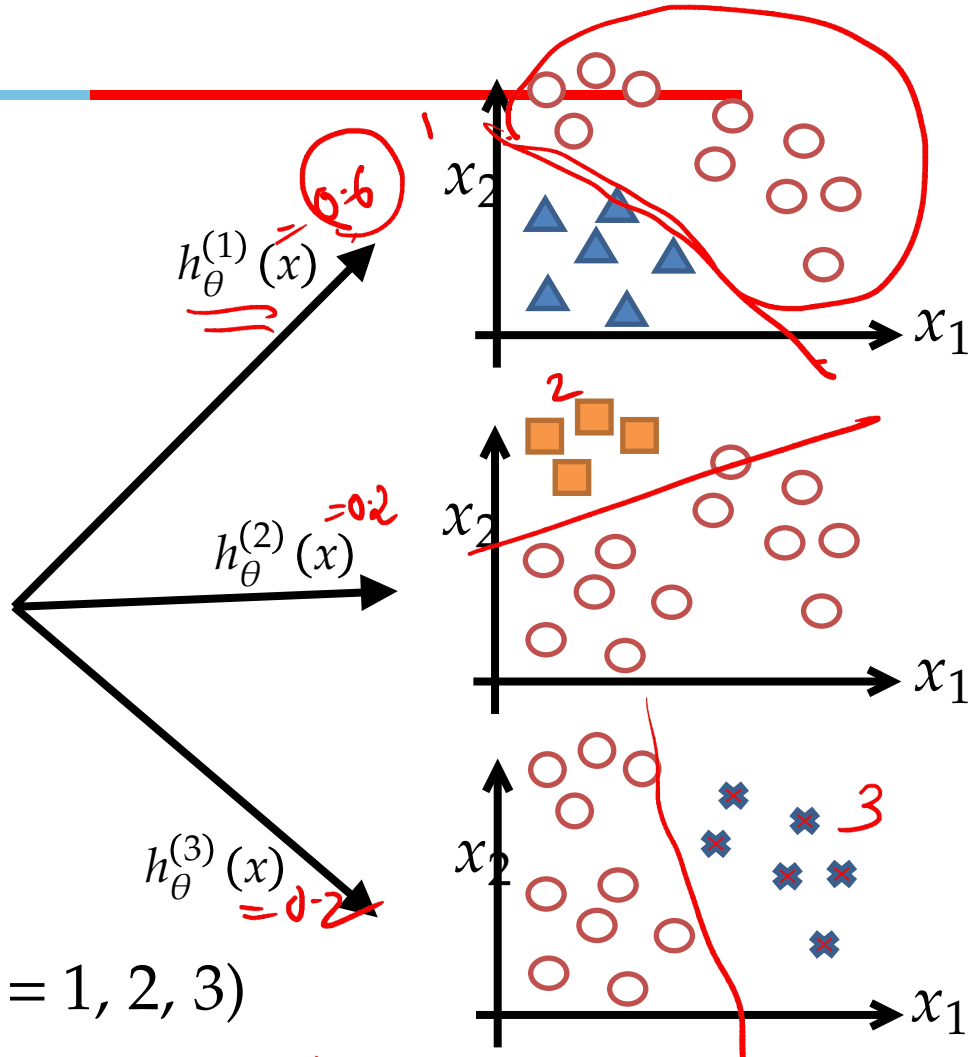
One-vs-all (one-vs-rest)



Class 1:

Class 2:

Class 3:



$$h_\theta^{(i)}(x) = P(y = i | x; \theta) \quad (i = 1, 2, 3)$$

$\Rightarrow P(y=1|x;\theta)$, $P(y=2|x;\theta)$, $P(y=3|x;\theta)$

Slide credit: Andrew Ng

Logistic regression (Classification)

- **Model**

$$h_\theta(x) = P(Y=1|X_1, X_2, \cdots, X_n) = \frac{1}{1+e^{-\theta^\top x}}$$

- **Cost function**

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

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

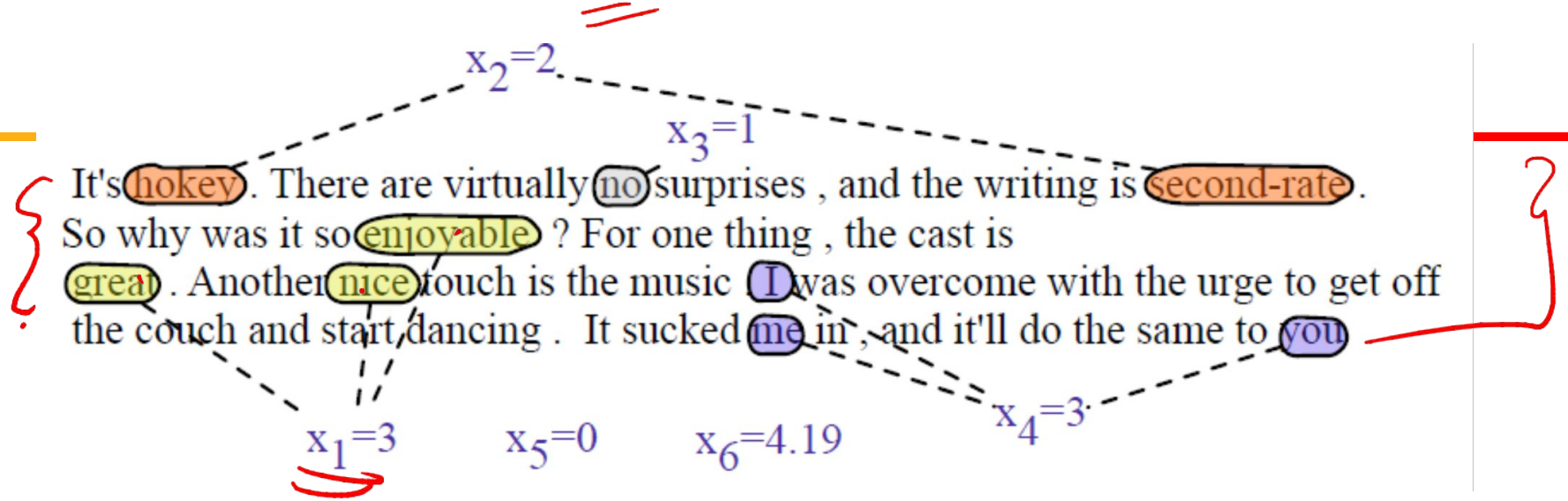
- **Learning**

Gradient descent: Repeat $\{\theta_j := \theta_j - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}\}$

- **Inference**

$$\hat{Y} = h_\theta(x^{\text{test}}) = \frac{1}{1+e^{-\theta^\top x^{\text{test}}}}$$

Example: Sentiment Analysis



Var	Definition	Value in Fig. 5.2
x_1	count(positive lexicon) $\in$ doc)	3
x_2	count(negative lexicon) $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if "no"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(66) = 4.19$

	x_1	x_2	x_3	x_4	x_5	x_6	y
①	3	2	1	3	0	4.19	
②							
③							

Sentiment Features

Classifying sentiment using logistic regression

Suppose w = $[2.5, -5.0, -1.2, 0.5, 2.0, 0.7]$

b = 0.1

$$
\begin{aligned}
p(+|x) = P(Y=1|x) &= \sigma(w \cdot x + b) \\
&= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\
&= \sigma(.833) \\
&= 0.70
\end{aligned}
$$

$$
\begin{aligned}
p(-|x) = P(Y=0|x) &= 1 - \sigma(w \cdot x + b) \\
&= 0.30
\end{aligned}
$$

Logistic Regression – Fit a Model

CGPA	IQ	Job Offered
5.5	6.7	1
5	7	0
8	6	1
9	7	1
6	8	0
7.5	7.3	0

Hyper parameters:

Learning Rate = 0.3

Initial Weights = (0.5, 0.5,0.5)

Regularization Constant = 0

$$\theta_0 := \theta_0 - 0.3\frac{1}{6}\sum_{i=1}^{6}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)(1)$$

$$\theta_{CGPA} := \theta_{CGPA} - 0.3\frac{1}{6}\sum_{i=1}^{6}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}_{CGPA}$$

$$\theta_{IQ} := \theta_{IQ} - 0.3\frac{1}{6}\sum_{i=1}^{6}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}_{IQ}$$

$$\theta^T X = 0.5 + 0.5\ CGPA + 0.5\ IQ$$

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

$$h_\theta(x) = \frac{1}{1 + e^{-(0.5 + 0.5\ CGPA + 0.5\ IQ)}}$$

Approx. : New weights

$\theta_0 = 0.4$

$\theta_{1=CGPA} = -0.4$

$\theta_{2=IQ} = -0.6$

Logistic Regression – Inference & Interpretation

CGPA	IQ	Job Offered
5.5	6.7	1
5	7	0
8	6	1
9	7	1
6	8	0
7.5	7.3	0

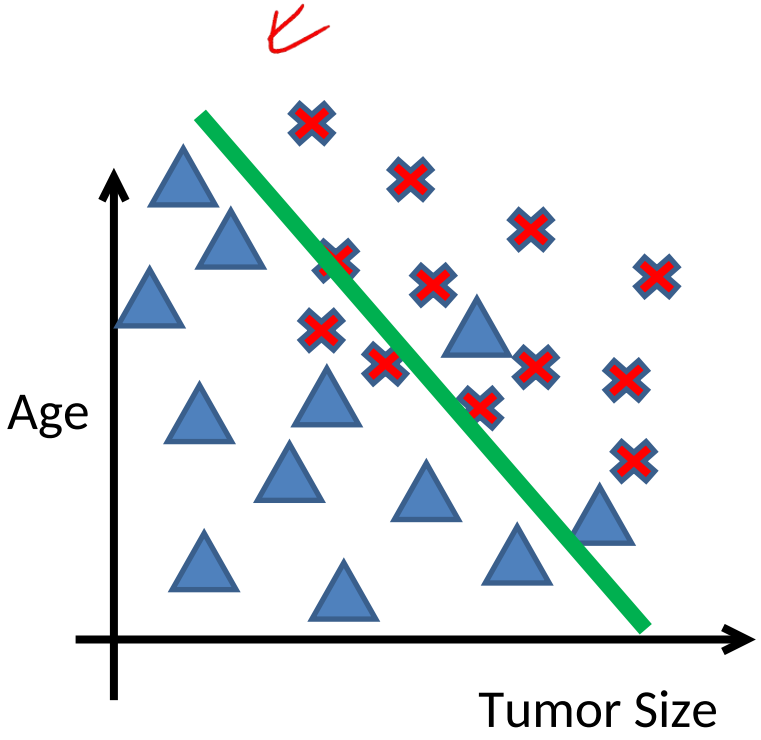
Assume : $0.4+0.3\text{CGPA}-0.45\text{IQ}$

Predict the Job offered for a candidate : (5, 6)

$h(x) = 0.31$

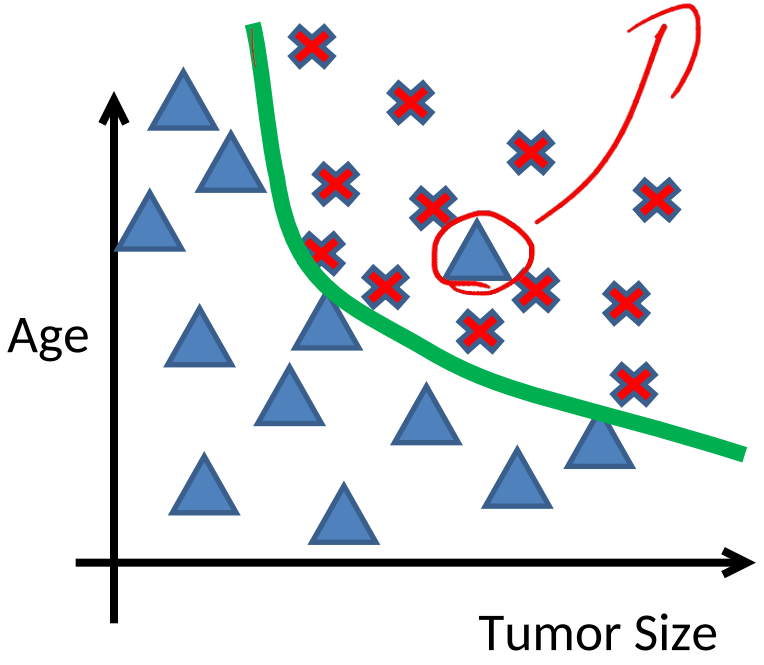
Y-Predicted = 0 / No

Logistic regression - overfitting

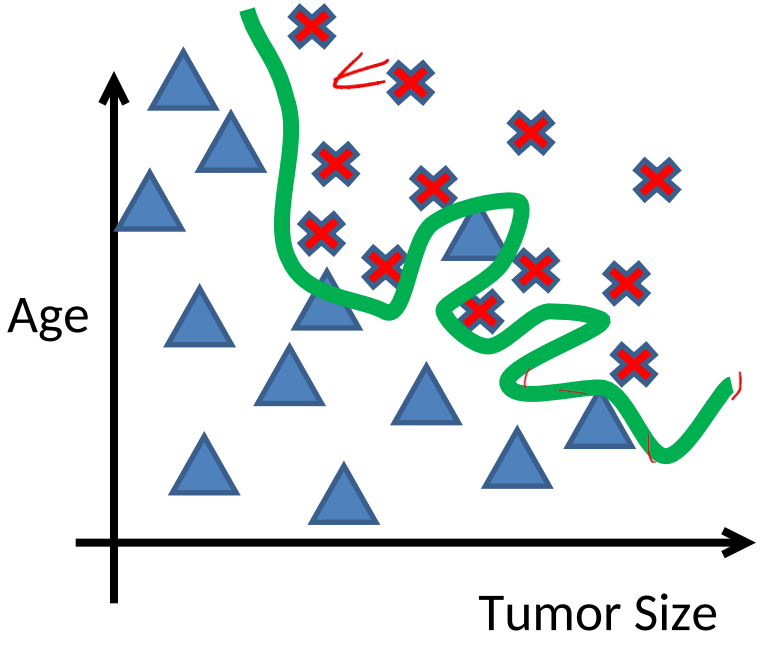


$$h_\theta(x) = g(\theta_0 + \theta_1 x + \theta_2 x_2)$$

Underfitting



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

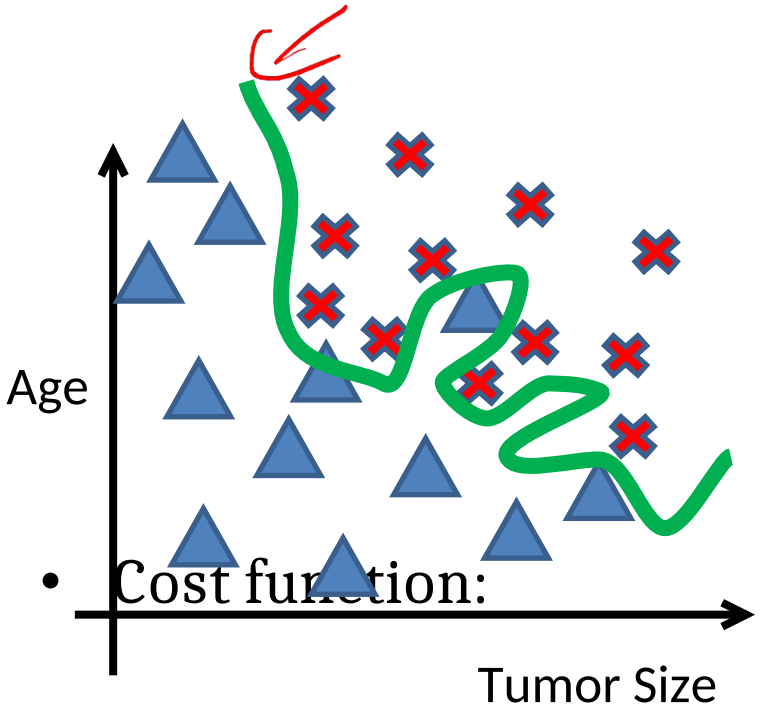


$$h_\theta(x) = g(\theta_0 + \theta_1 x + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 + \theta_6 x_1^3 x_2 + \theta_7 x_1 x_2^3 + \cdots)$$

Overfitting

Slide credit: Andrew Ng

Regularized logistic regression



$$h_\theta(x) = g(\theta_0 + \theta_1 x + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 + \theta_6 x_1^3 x_2 + \theta_7 x_1 x_2^3 + \cdots)$$

- Cost function:

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

Slide credit: Andrew Ng

Regularized logistic regression

Regularized cost function (m=training examples, n = features)

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Gradient descent with regularized cost function

Repeat {

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

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

}

Just like with linear regression, we will want to **separately** update θ_0 and the rest of the parameters because we do not want to regularize θ_0.

References

- Tom M. Mitchell
  Generative and discriminative classifiers: Naïve Bayes and Logistic Regression
  http://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf
- https://cs229.stanford.edu/lectures-spring2022/main_notes.pdf
- Pattern recognition and machine learning, Christopher bishop – CH 1.5
- http://papers.nips.cc/paper/2020-on-discriminative-vs-generative-classifiers-a-comparison-of-logistic-regression-and-naive-bayes.pdf

Self Learning

Logistic regression GD derivation

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

$$\sigma(\mathrm{x}) = \frac{1}{1 + e^{-\mathrm{x}}}$$

$$\frac{d\big(\sigma(\mathrm{x})\big)}{dx} = \frac{0 * (1 + e^{-\mathrm{x}}) - (1) * \big(e^{-\mathrm{x}} * (-1)\big)}{(1 + e^{-\mathrm{x}})^2}$$

$$\frac{d\big(\sigma(\mathrm{x})\big)}{dx} = \frac{(e^{-\mathrm{x}})}{(1 + e^{-\mathrm{x}})^2} = \frac{1\text{-}1 + (e^{-\mathrm{x}})}{(1 + e^{-\mathrm{x}})^2} = \frac{1 + e^{-\mathrm{x}}}{(1 + e^{-\mathrm{x}})^2} - \frac{1}{(1 + e^{-\mathrm{x}})^2}$$

$$\frac{d\big(\sigma(\mathrm{x})\big)}{dx} = \frac{1}{1 + e^{-\mathrm{x}}} * \left(1 - \frac{1}{1 + e^{-\mathrm{x}}}\right) = \sigma(\mathrm{x})\big(1 - \sigma(\mathrm{x})\big)$$

Logistic regression cost function

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

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

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

Slide credit: Andrew Ng

Step –I

Applying Chain rule and writing in terms of partial derivatives

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

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

$$\frac{\partial(J(\theta))}{\partial(\theta j)} = -\frac{1}{m} * \left(\sum_{i=1}^{m} \left[y^{(i)} * \frac{1}{h_\theta(x^{(i)})} * \sigma(z)\left(1 - \sigma(z)\right) * \frac{\partial(\theta^T x)}{\partial(\theta j)} \right]\right.$$

$$\left. + \sum_{i=1}^{m} \left[\left(1 - y^{(i)}\right) * \frac{1}{\left(1 - h_\theta(x^{(i)})\right)} * \left(-\sigma(z)\left(1 - \sigma(z)\right) * \frac{\partial(\theta^T x)}{\partial(\theta j)} \right] \right)$$

Step –II

- Evaluating the partial derivative using the pattern of the derivative of the sigmoid function.

$$\frac{\partial(J(\theta))}{\partial(\theta j)} = -\frac{1}{m}*(\sum_{i=1}^{m}\left[y^{(i)} * \frac{1}{h_\theta(x^{(i)})} * \sigma(z)(1-\sigma(z)) * \frac{\partial(\theta^T x)}{\partial(\theta j)}\right]$$

$$+\sum_{i=1}^{m}\left[(1-y^{(i)}) * \frac{1}{(1-h_\theta(x^{(i)}))} * (-\sigma(z)(1-\sigma(z))* \frac{\partial(\theta^T x)}{\partial(\theta j)}\right])$$

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

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

Step –III

- Simplifying the terms by multiplication

$$\frac{\partial(J(\theta))}{\partial(\theta j)} = -\frac{1}{m} * \left(\sum_{i=1}^{m} \left[y^{(i)} * \left(1 - h_\theta\left(x^{(i)}\right)\right) * x_j^i - \left(1 - y^{(i)}\right) * h_\theta\left(x^{(i)}\right) * * x_j^i \right]\right.$$

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

$$\frac{\partial(J(\theta))}{\partial(\theta j)} = -\frac{1}{m} * \left(\sum_{i=1}^{m} \left[y^{(i)} - h_\theta\left(x^{(i)}\right)\right] * x_j^i \right)$$