Machine Learning with R

Mohamed Rahouti

Outlines

- Setups
- Introduction to machine learning
- Machine learning algorithms
- Supervised learning
- Decision Trees
- Unsupervised learning
- Conclusions

Configs/Setups

- RStudio
- H2O
- Java environment (if not already installed in the system)
- ggplot or ggplot2 (depending on R version)

Introduction to Machine Learning (ML)

What is ML?

- Arthur Samuel (1959). Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed.

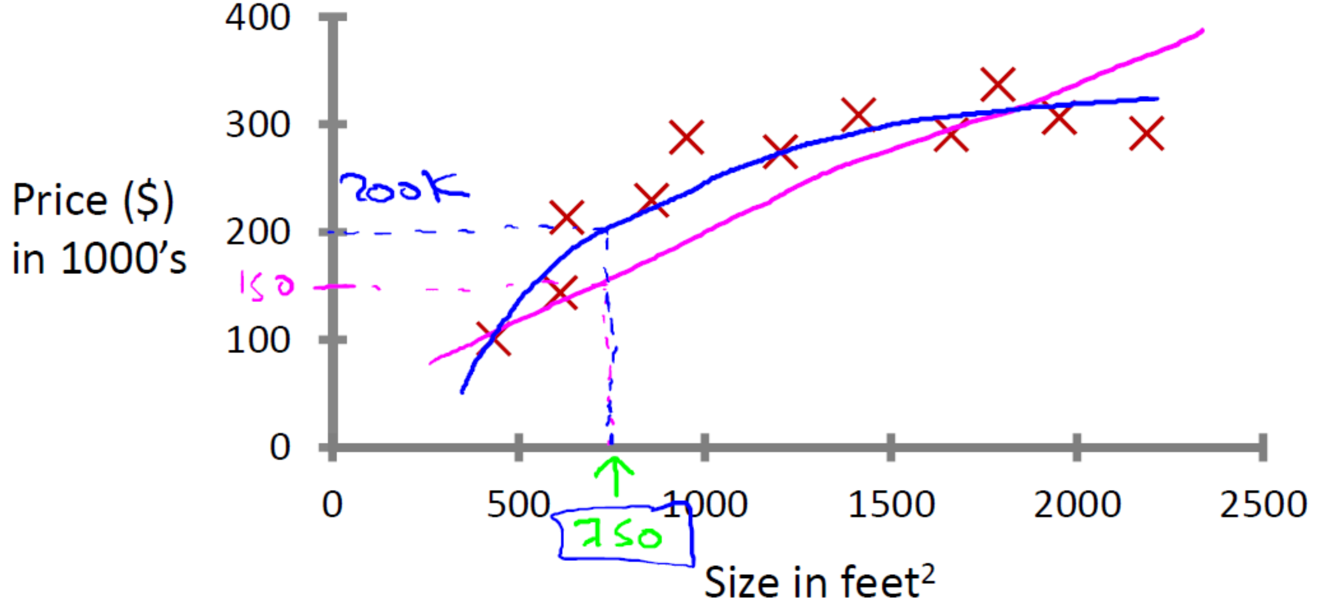
- Tom Mitchell (1998) Well-posed Learning Problem: A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

ML Algorithms

- Machine learning algorithms:
  - Supervised learning
  - Unsupervised learning
- Others: Reinforcement learning, recommender systems.
- Also talk about: Practical advice for applying learning algorithms.

Supervised Learning

- Prediction of house pricing



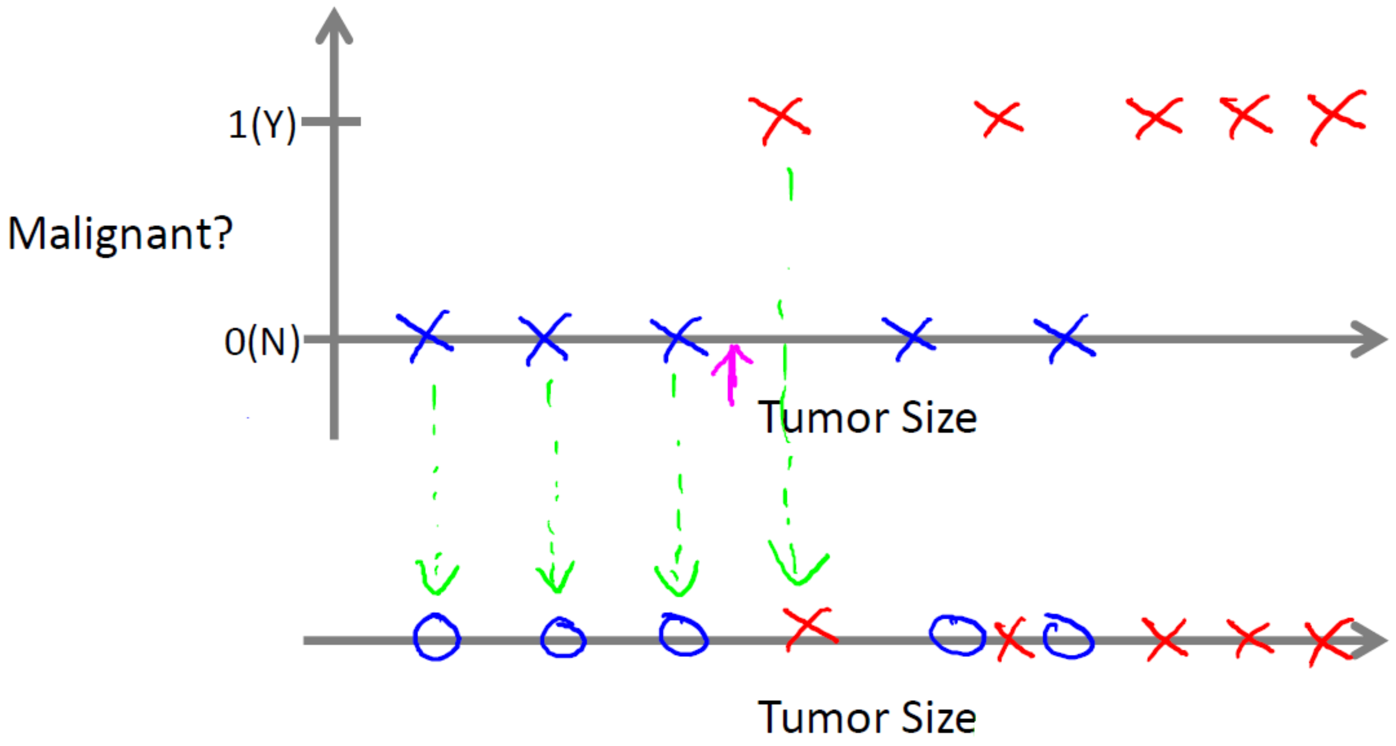
<u>Supervised Learning</u>

' right answers' given

<u>Regression</u>: Predict continuous valued output (price)

Supervised Learning

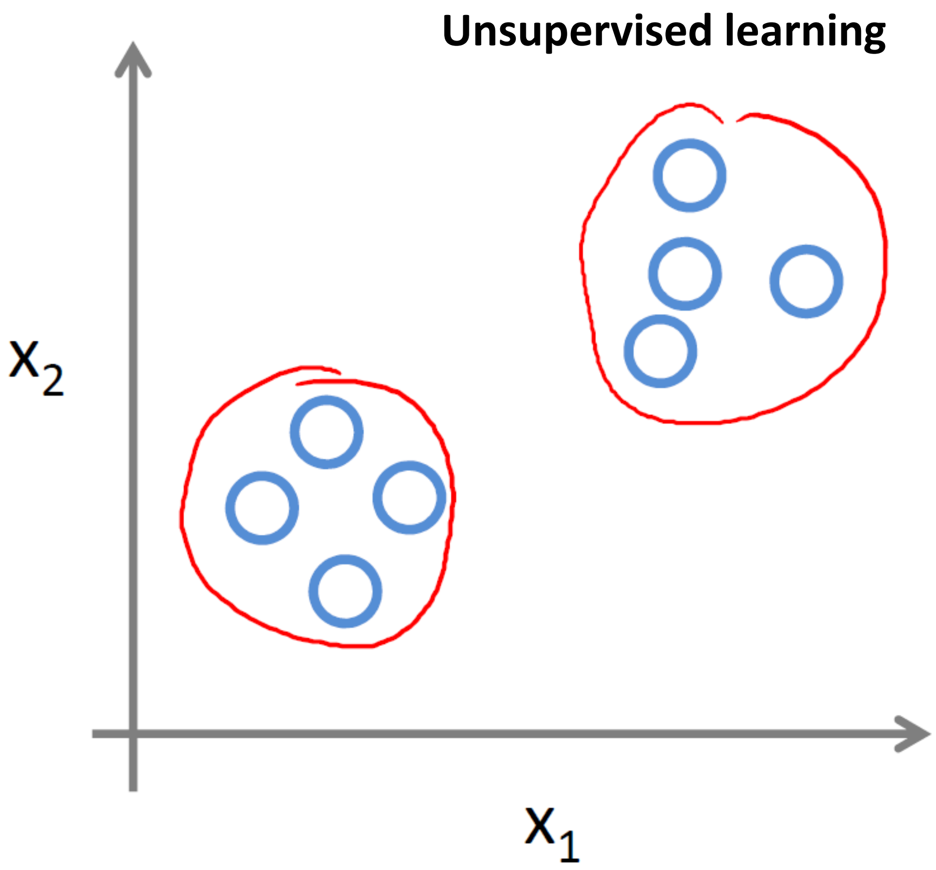
- Prediction of breast cancer (malignant, benign)

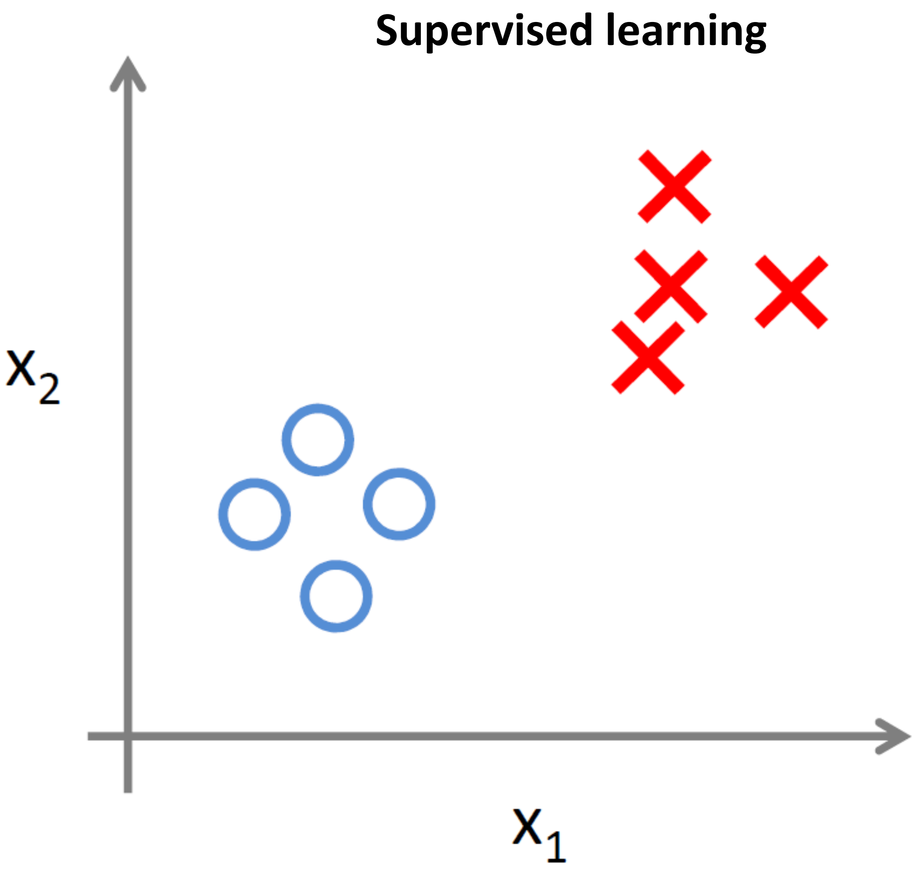


Classification

Discrete valued output (0 or 1)

Unsupervised Learning





Supervised Learning (1): Linear Regression

- Training set of housing prices

- (x, y) = one training example
- (x^i, y^i) = *ith* training example

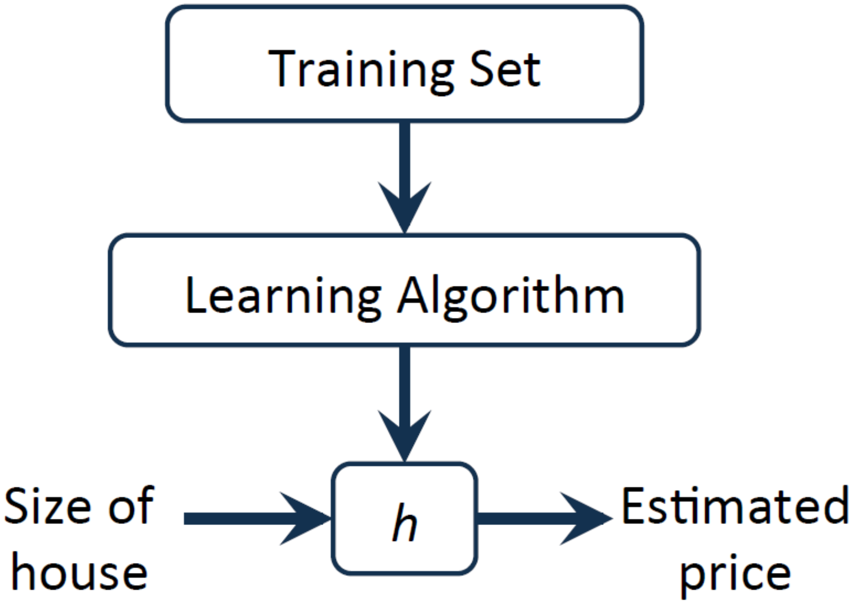
Size in feet² (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

Notation:

m = Number of training examples
x's = "input" variable / features
y's = "output" variable / "target" variable

Linear Regression: Model Representation

- How to represent H?



$h_\theta(x) = \theta_0 + \theta_1 x$

Shorthand: $h(x)$

$h(x) = \theta_0 + \theta_1 x$

Linear Regression: Cost Function

- Training Set

Size in feet^2 (x)	**Price ($) in 1000's (y)**
2104	460
1416	232
1534	315
852	178
…	…

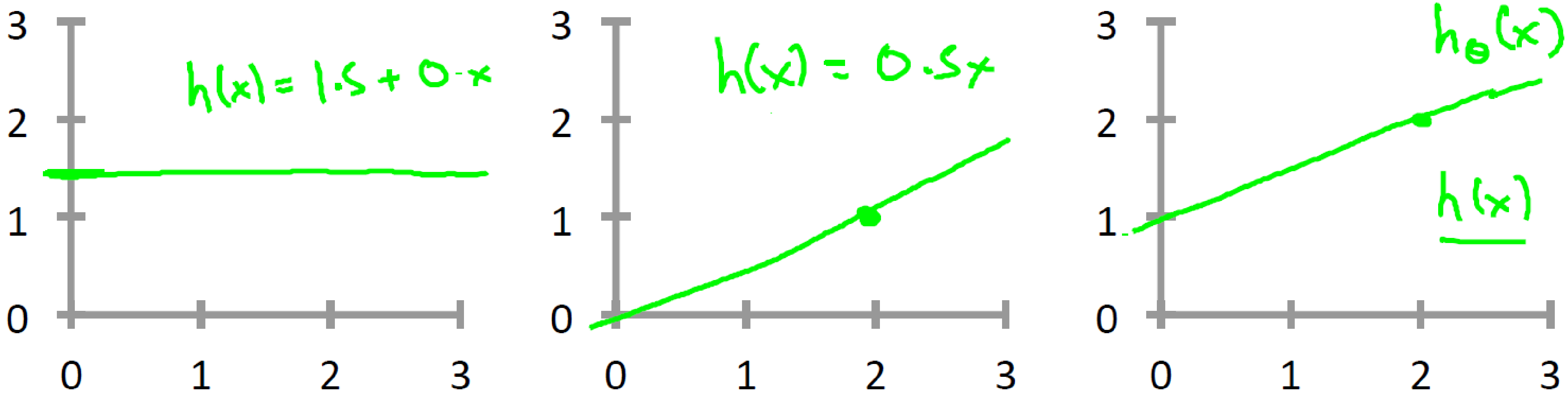
Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

θ_i's: Parameters

- But how to choose θ_i's ?

Linear Regression: Cost Function (Cont.)

$$h_\theta(x) = \theta_0 + \theta_1 x$$

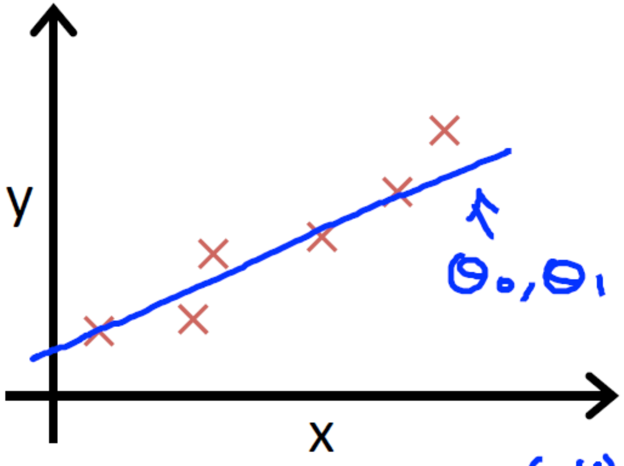


$\theta_0 = 1.5$
$\theta_1 = 0$

$\theta_0 = 0$
$\theta_1 = 0.5$

$\theta_0 = 1$
$\theta_1 = 0.5$

Linear Regression: Cost Function (Cont.)



θ_0, θ_1

$(x^{(i)}, y^{(i)})$

Idea: Choose θ_0, θ_1 so that $h_\theta(x)$ is close to y for our training examples (x, y)

\# training examples

$$\boxed{\underset{\theta_0 \theta_1}{\text{minimize}}} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$h_\theta(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$\underset{\theta_0, \theta_1}{\text{minimize}} \underbrace{J(\theta_0, \theta_1)}_{\text{Cost function}}$$

Squared error function

Logistic Regression: Cost Function (Cont)

- Again,

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters: θ_0, θ_1

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Linear Regression: Gradient Descent

Have some function $J(\theta_0, \theta_1)$

Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Outline:

- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$

  until we hopefully end up at a minimum

Linear Regression: Gradient Descent (Cont.)

Have some function $J(\theta_0, \theta_1)$

Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Outline:

- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$

  until we hopefully end up at a minimum

Gradient descent algorithm

$$
\begin{aligned}
&\text{repeat until convergence } \{ \\
&\quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \\
&\qquad (\text{for } j = 1 \text{ and } j = 0) \\
&\}
\end{aligned}
$$

Linear Regression Model

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Application in R

- Let's implement our function called **gradientR** as follows

```
gradientR<-function(y, X, epsilon,eta, iters){
      epsilon = 0.0001
      X = as.matrix(data.frame(rep(1,length(y)),X))
      N= dim(X)[1]
      print("Initialize parameters...")
      theta.init = as.matrix(rnorm(n=dim(X)[2], mean=0,sd = 1)) # Initialize theta
      theta.init = t(theta.init)
       e = t(y) - theta.init%*%t(X)
       grad.init = -(2/N)%*%(e)%*%X
       theta = theta.init - eta*(1/N)*grad.init
       l2loss = c()
      for(i in 1:iters){
          l2loss = c(l2loss,sqrt(sum((t(y) - theta%*%t(X))^2)))
          e = t(y) - theta%*%t(X)
          grad = -(2/N)%*%e%*%X
          theta = theta - eta*(2/N)*grad
            if(sqrt(sum(grad^2)) <= epsilon){
              break
            }
        }
  print("Algorithm converged")
  print(paste("Final gradient norm is",sqrt(sum(grad^2))))
  values<-list("coef" = t(theta), "l2loss" = l2loss)
  return(values)
}
```

Application in R (Cont.)

- Let's also make a function that estimates the parameters with the normal equations:

$$\theta = (X^T X)^{-1} X^T Y$$

```
normalest <- function(y, X){
    X = data.frame(rep(1,length(y)),X)
    X = as.matrix(X)
    theta = solve(t(X)%*%X)%*%t(X)%*%y
    return(theta)
}
```

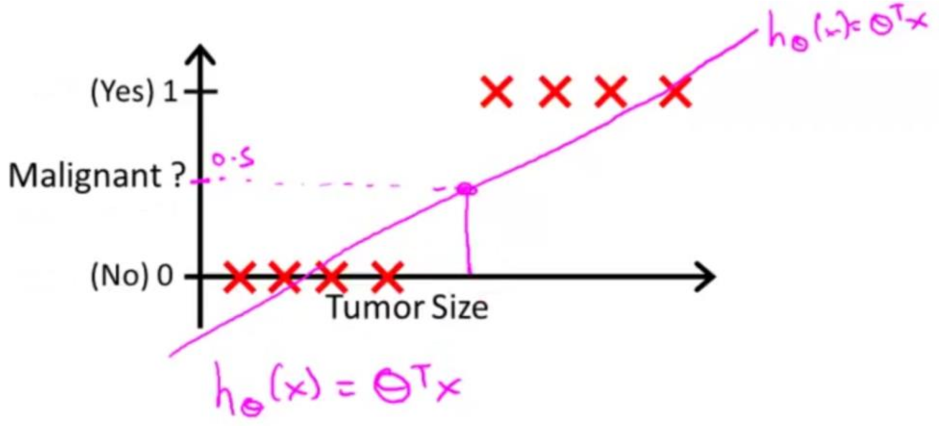
Logistic Regression: Classification

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)



Classification: y = 0 or 1

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

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

Threshold classifier output $h_\theta(x)$ at 0.5:

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

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

Logistic Regression: Model Representation

Want $0 \le h_\theta(x) \le 1$

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

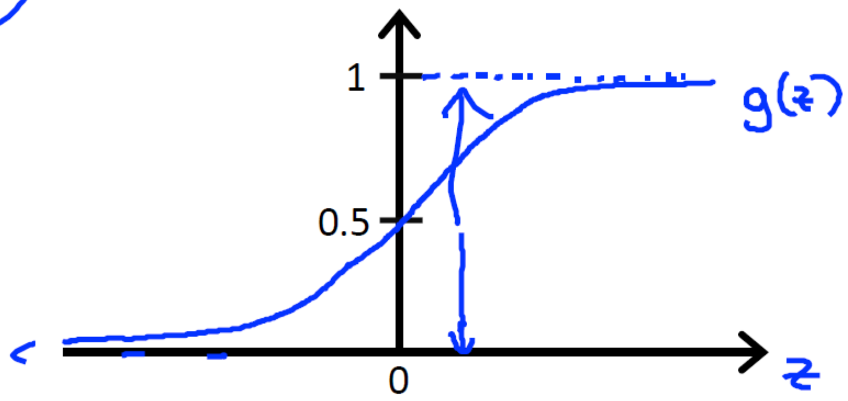
$$g(z) = \frac{1}{1+e^{-z}}$$

$\theta^T x$

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

Sigmoid function

Logistic function



Parameters θ

Logistic Regression: Interpretation of Hypothesis Testing

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

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

$h_\theta(x) = 0.7$ $y=1$

Tell patient that 70% chance of tumor being malignant

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

$y = 0$ or 1

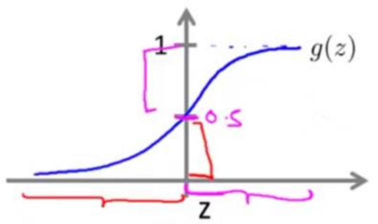
"probability that y = 1, given x, parameterized by θ"

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

Logistic Regression: Decision Boundary

$h_\theta(x) = g(\theta^T x)$ = $P(y=1 \mid x; \theta)$

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



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

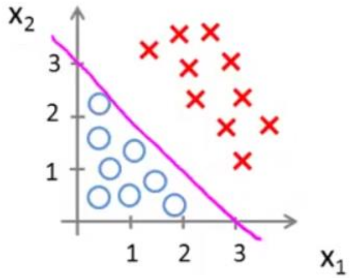
$\theta^T x \geq 0$

$g(z) \geq 0.5$
when $z \geq 0$
$h_\theta(x) = g(\theta^T x) \geq 0.5$
whenever $\theta^T x \geq 0$
z

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

$h_\theta(x) = g(\theta^T x)$
$\theta^T x < 0$

$g(z) < 0.5$



$\theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$

$h_\theta(x) = g(\underbrace{\theta_0}_{=-3} + \underbrace{\theta_1}_{=1} x_1 + \underbrace{\theta_2}_{=1} x_2)$

Predict "$y = 1$" if $\underbrace{-3 + x_1 + x_2}_{\theta^T x} \geq 0$

$\rightarrow x_1 + x_2 \geq 3$

Logistic Regression: Cost Function

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

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

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

How to choose parameters θ ?

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

Logistic Regression: 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)

Application in R

- Lab 2

Decision Tree

- Graph G to represent choices & their results in form of a tree.
  - Nodes in G represent an event or choice
  - Edges in graph represent the decision rules or conditions
- Example:
  - Predicting an email as spam or not spam
  - Predicting of a tumor is cancerous
  - Predicting a loan as a good or bad credit risk based on the factors in each of these
- A model is created with observed data also called training data
- A set of validation data is used to verify and improve the model

Decision Tree in R

- R has packages which are used to create and visualize decision trees.
- For new set of predictor variable, we use this model to arrive at a decision on the category (yes/No, spam/not spam) of the data.
- The R package **"party"** is used to create decision trees.
- The basic syntax for creating a decision tree in R
  - *> ctree(formula, data)*
  - **formula** is a formula describing the predictor and response variables.
  - **data** is the name of the data set used.

Application in R

- Lab 3

Thanks for your Attention

- Questions??

Credits

- Some material was borrowed from:
  - Machine Learning with R and H2O
  - Introduction to ML with Applications in R
  - ML in R, by Alexandros Karatzoglou
  - ML course, by A. NG