No.

Date

Machine Learning

Introduction:

(web search, photo tagging, ~~email~~ anti-spam)

Machine learning —— example : search engine, photo recognisation, spam filter in email.

↳ mimic algorithm to know how human learn

~~(most recent stages of development)~~

⇒ to learn state-of-the-art ML algorithm

⇒ to know how to get this stuff work on problem you care about

ML - grown out of field of AI

↳ ~~ca~~ new capability for computers and touch a lot of things in science

Learning algorithm

Example : 1. Database mining

— Large ~~databa~~ datasets from growth of automation/web

e.g. - web click ~~data~~ data ~~etc~~ (clickstream data), medical record,

collecting ~~data so to~~ understand user/human better

2. Applications can't program by hand

Eg. - Autonomous helicopter, handwriting recognition, most of Natural Language Processing (NLP), Computer Vision

we can't ~~write the~~ helicopter program, but ~~write~~ to let computer understand and learn by itself to fly

3. Self-customising programs

Eg. - Amazon, Netflix product recommendations

4. Understanding human learning (brain, real AI)

No.

Date

What is Machine Learning?

2 definitions

i) The field of study that gives computer the ability to learn without being explicitly programmed.

ii) A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as as measured by P, improves with experience E.

eg.: Playing checkers

E = experience of playing ~~g~~many games of checkers

T = task of playing checkers

P = the probability that program will win next game.

Machine learning problem be assigned to one of two broad classification;

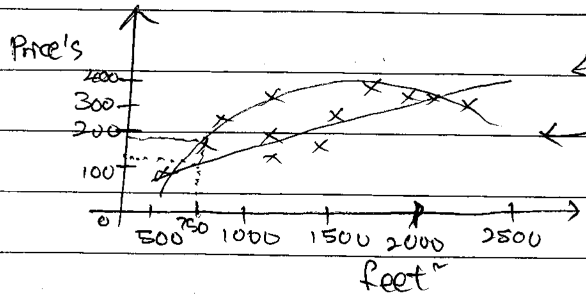
~~Sup~~ - Supervised learning and Unsupervised learning

Date

Supervised Learning

Example :

Housing price prediction ;



Can use a computer algorithm to draw straight line to check the price of 750 feet²

or even make a quadratic function to the data .

So, to discuss whether use straight line or quadratic function

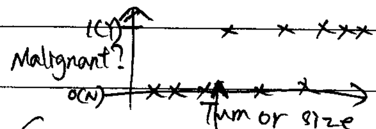
Supervised Learning can be said as — give right answer to computer.
↳ give ~~a~~ right data set how much houses are sell
→ task of algorithm is to produce more right answer

in this case

in term ⇒ Regression = Predict continuous value output (price)

Example :

Breast cancer (malignant , benign)



~~If patient size is here, machine learning question is to estimate the~~
probability / chance for tumor is malignant or benign .

Kind of classification problem

Discrete valued output (0 or 1)

⇒ trying to predict 0 or 1 output.

In this problem can have more than 2 values for the two possible value for output.

or there's other type of breast cancer, it will be 0, 1, 2, 3

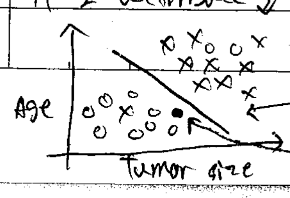
no, first type, second, third

Another way plot data

Benign use "o" and malignant remain "x"

~~o o o x o x o x x x~~
~~Tumor size (only 1 attribute)~~

if 2 attribute



Learning algorithm ~~with~~ will draw a straight line to separate 2 classes of algorithm.

— if someone falls in here, ~~me~~ will predict most likely is benign.

No.

Date

Learning algorithm can take infinite features/cues/attributes to make prediction. But how?

⇒ An algorithm, Support Vector Machine
↳ a neat mathematical trick allow computer to deal with infinite number of features.

Conclusion:

Supervised learning, ~~in~~ in our data set, we are told what is the correct answer that we would have quite liked the algorithms have predicted on that example.

说那么明显 → By regression, our goal is to predict a continuous valued output.

分得 → classification problem, the goal is to predict a discrete value output

很明显 自放进几个 class 来 predict problem 在哪一个 class

Supervised learning — we are given a data set and already know what our correct output should look like, having the idea that there is a relationship between input and output.

↓ Problems

regression → to predict within a continuous output, we are trying to map input variables to some continuous function

Classification → we are instead trying to predict results in a discrete output. we are trying to map input variables into discrete categories.

Housing price prediction ⇒ Given data of houses on real estate market, try to predict their price, price as a function of size is a continuous output, so is a regression problem
- if we turn this into making our output about whether the house "sells for more or less than asking price" Then we are classifying houses based on price into two discrete categories.

B/M Tumor prediction — Regression — Given picture of person, we have to predict their age on the basis of given picture.

Classification — Given patient with a tumor, we have to predict the tumor is malignant or benign.

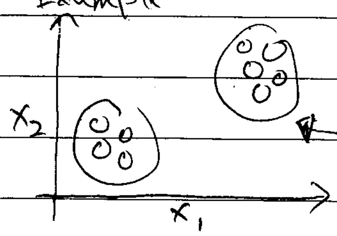
labelled data is like "spam" and "not spam" 已经给答案.

Unsupervised Learning

We are given a data set, and not given any label on the data, and not told ~~t~~ what to do and what each data point is.

We just have to find out the structure of dataset.

Example



(all same)

Given a data set, unsupervised learning algorithm might decide data lives in 2 diff. clusters, will break these data into 2 separate clusters.

∴ clustering algorithm

example: Google news

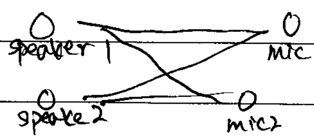
↳ 一个标题 news, online 有很多不同的 stories, google cluster 他们全部在一个大标题

Used for other application:

⇒ - Organize large computer clusters (figure out which machines tend to work together, put them tgt, so work better)

- Social network analysis (identify the cohesive groups of friends)
- Market segmentation (group the customers)
- Astronomical data analysis (how galaxies are formed)

Cocktail party problem



Both of speakers speak at the same time but microphone not receiving the same time, so algorithm can separate out the 2 voices of speakers by using Octave type programming environment

---

To approach problems with little or no idea what results should look like, we derive structure from dataset

— by clustering the data based on relationships among variables in data with unsupervised learning, no feedback based on ~~per~~ prediction results.

example:

clustering: Take collection of ~~to~~ 1 million diff. genes, group gene into groups that are somehow similar or related by different variables, such as lifespan, location, roles ...... and so on.

Non-clustering: The "Cocktail Party algorithm", allow you to find structure in a chaotic environment. (identify individual voices and music from a mesh of sounds at a cocktail party).

Model Representation

example

Training set of housing prices

Size in feet^2 (x)	Price in 1000's (y)
2104	460
1234	232
⋮	⋮

$m = 47$ example

m = Number of training examples

x = "input" variable / features

y = "output" variable / "target" variable

(x, y) = one training example

$(x^{(i)}, y^{(i)})$ — i^{th} training example

$x^1 = 2104$ $\quad y^1 = 460$

$x^2 = 1234$ $\quad y^2 = 232$

$(x^{(i)}, y^{(i)})$; $i = 1, \dots, m$ — is called a training set.

use X to denote the space of input values and Y to denote space of output values.

function $h: X \rightarrow Y$, ~~to~~ make $h(x)$ is a good "predictor" for corresponding value of y "h" can be called a hypothesis

Training set

↓

Learning algorithm

↓

$x \rightarrow \boxed{h} \rightarrow$ predicted y

(living area of house) (predicted price of house)

When target variable we trying to predict is continuous, such as housing example, problem is a regression problem, when y take on only a small number of discrete value, (if giving living rea, we predict it is a house or apartment), is a ~~a~~ classification problem.

Date

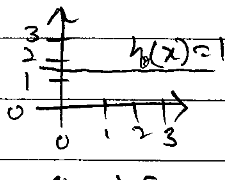
Cost Function

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

θ_i's : Parameters

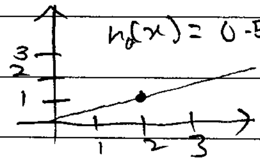
How to choose ~~θ_i's~~ θ_i's ?

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



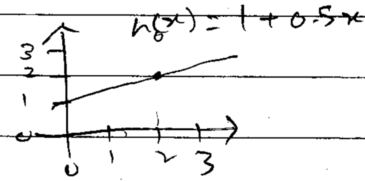
$\theta_0 = 1.5$

$\theta_1 = 0$



$\theta_0 = 0$

$\theta_1 = 0.5$



$\theta_0 = 1$

$\theta_1 = 0.5$

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

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

$m \rightarrow$ # of training set

$h_\theta(x^{(i)})$: hypothesis of prediction of house price

$y^{(i)}$: actual house price

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

$$\underset{\theta_0, \theta_1}{\text{minimize}} \underbrace{J(\theta_0, \theta_1)}_{\text{cost function / Squared error function}}$$

- Measure the accuracy of our hypothesis function using cost function
- This takes an average difference of all results of hypothesis with input x and actual output y. $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$
- To break it apart, it is $\frac{1}{2}\bar{x}$ where $\bar{x}$ is the mean of square of $h_\theta(x_i) - y_i$, or difference between predicted value and actual value.
- The function is otherwise called "Square error function" or "Mean square error". Mean is halved $(\frac{1}{2})$ as convenience for computation of gradient descent, as the derivative term of square function will cancel out $\frac{1}{2}$ term.

Cost Function - Intuition 1

Hypothesis:

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

Parameters:

θ_0, θ_1

Cost Function:

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

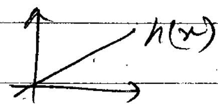
Goal: $\underset{\theta_0, \theta_1}{\text{minimize}} \; J(\theta_0, \theta_1)$

simplified

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

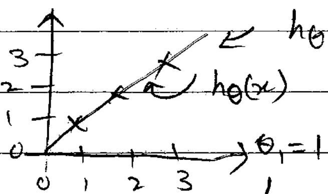
$\theta_0 = 0$

θ_1



$h_\theta(x)$

for fixed θ, this is a function of x



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

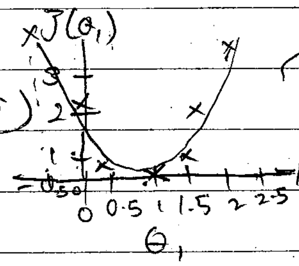
$$= \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_1 x^{(i)} - y^{(i)} \right)^2$$ 有几个就加几个，这里有 $m=3$，所以加3个

$$= \frac{1}{2m} (0^2 + 0^2 + 0^2) = 0$$

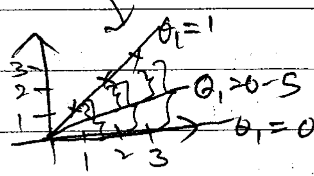
$J(1) = 0$

$J(\theta_1)$

(function of parameter θ_1)



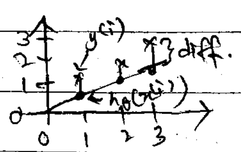
Keep increasing when leaving further from the line



$\theta_1 = 1$ = global minimum.

choose the value of θ_1 that can bestly is to minimize $J(\theta_1)$

When graph go to $\theta_1 = 0.5$



$$J(0.5) = \frac{1}{2m} \left[(0.5-1)^2 + (1-2)^2 + (1.5-3)^2 \right]$$

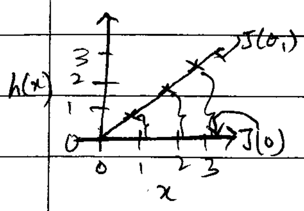
$$= \frac{1}{6}(3.5) \approx 0.58$$

No. ______

Date ______

Example:

Training set $m=3$, hypothesis representation is $h_\theta(x) = \theta_1 x$ with parameter θ_1.
The cost function $J(\theta_1)$ is $J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$. What is $J(0)$?



$$J(0) = \frac{1}{2m}\left((0-1)^2 + (0-2)^2 + (0-3)^2\right)$$
$$= \frac{14}{6}$$

Objective ⇒ get best possible line

↳ will be such so that average squared vertical distance of scatter points from the line will be the least. Ideally, line should pass through all points of our training data set. In such a case, value of $J(\theta_0, \theta_1)$ will be 0. The ~~following example~~

Linear Regression with one variable   Cost function intuition II (contour plot) 3-D

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

Gradient descent ⇒ minimize cost function J by minimizing each θ (minus α(derivative of $J(\theta)$) using slope to reach minimum value of θ are to minimum value of $J(\theta)$)

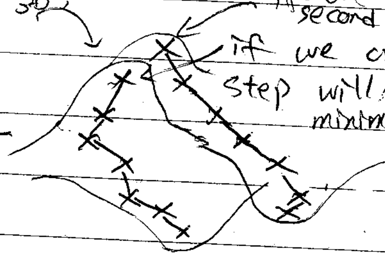
Have some function $J(\theta_0, \theta_1) \Rightarrow J(\theta_0, \theta_1, \theta_2, \dots \theta_n)$

Want to $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1) \Rightarrow \min_{\theta_0 \dots \theta_n} J(\theta_0, \dots \theta_n)$

Outline:

- Start with some θ_0, θ_1 (initial guess) ⇒ (say $\theta_0 = 0, \theta_1 = 0$)
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ → keep changing until we hopefully end up at a minimum

Example: ~~Contour Plot~~



if at other point, will go to second local optimum

if we on this point, which step will we take to reach minimum faster

Gradient descent algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \boxed{\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)} \quad (\text{for } j = 0 \text{ and } j = 1)$$

}

Update (assign new variable)

learning rate (control how big a step we take downhill with creating descent.) $\alpha \uparrow$, step ↑

derivative

to simultaneously update θ_0 and θ_1

[j = 0, 1 represents the feature index number]

update new a=b

Assignment	Truth assertion
a := b	a = b
a := a + 1 (increment)	a = a + 1 ✗

just declare a and b are same

Correct: Simultaneous update

$temp0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

$temp1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$

$\theta_0 := temp0$

$\theta_1 := temp1$

Incorrect:

$temp0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

$\theta_0 := temp0$

$temp1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$

Cannot because it will be updated first.

$\theta_1 := temp1$

The way we do is by taking derivative (the tangential line to a function) of our cost function. Slope of tangent is the derivative at the point and it will give us the direction to move towards.

Distance btwn each cross, "X" on the hill are determined by parameter a.

Direction in which step is taken is determined by partial derivative of $J(\theta_0, \theta_1)$. Depending on ~~which~~ where one starts on graph, one could end up at different points.

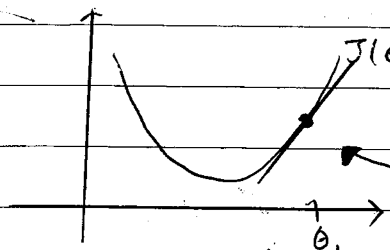
Gradient Descent Intuition

$\frac{\partial}{\partial \theta_1} \Rightarrow$ Partial derivative

$\frac{d}{d\theta_1} \Rightarrow$ derivative

depending on number of parameters in function J.

in this is the same thing



$(\theta_1 \in \mathbb{R})$

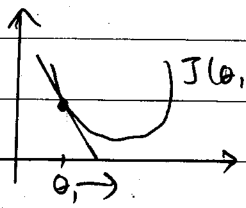
Gradient Descent will update ↓

$\theta_1 := \theta_1 - \alpha \boxed{\frac{d}{d\theta_1} J(\theta_1)}$

to find tangent line to function (slope) ≥ 0

decrease to get closer to min-point

$\theta_1 := \theta_1 - \alpha$ (+ve num.)



$\frac{d}{d\theta_1} J(\theta_1)$

going to be -ve num.

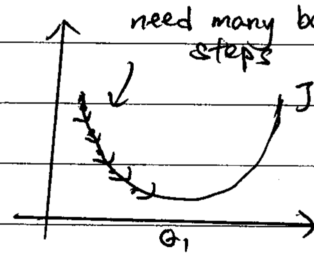
increase to get closer to min-point

$\theta_1 := \theta_1 - \alpha$ (-ve num)

$\theta_1 := \theta_1 + \alpha$ (num)

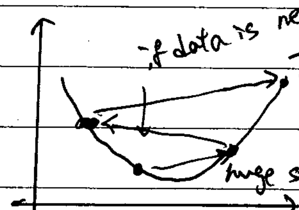
$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$

if α is too small, gradient descent can be slow to global minimum as need lot of steps.



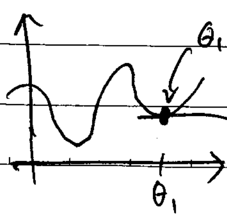
if α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.

不能 converge, 反而 diverge



huge step, then want to decrease will become it keeps getting further away from the minimum

Suppose θ_1 is at local optimum of $J(\theta_1)$, what will one step of gradient descent $\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$ do?



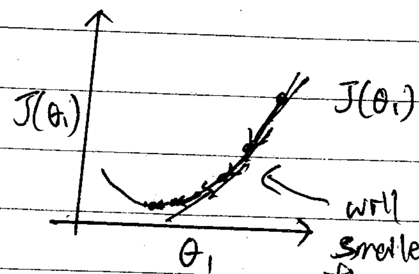
Answer: Will leave θ_1 unchanged

because ~~stp~~ slope of this line will become zero, thus derivative term is zero, gradient descent update ≠ zero · unchanged

And This is why gradient descent can converge to a local minimum, even with learning rate, α fixed.

$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$

As we approach a local minimum, gradient descent will automatically take smaller steps.
So, no need to decrease α over time.



will keep taking smaller step as the slope will decrease and converge to local min.

θ_1 减一个大的 slope, ~~变~~ θ_1 变小
θ_1 再减小一点的 slope, θ_1 变更小
θ_1 减更小的 slope, θ_1 变更更小

Linear regression with 1 variable

Gradient descent for linear regression

No.

Date

⇒ linear regression model

⇒ squared error cost function

Put together gradient descent ~~and~~ with cost function to give us an algorithm for linear regression or putting a straight line to our data

Gradient descent algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \boxed{\frac{\partial}{\partial\theta_j} J(\theta_0, \theta_1)}$$

(for $j=1$ and $j=0$)

}

key term

apply

minimize

Linear Regression Model

linear hypothesis

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

squared error cost function

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

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

$$= \frac{\partial}{\partial\theta_j}\frac{1}{2m}\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)^2$$

$$\theta_0\ j=0 : \frac{\partial}{\partial\theta_0} J(\theta_0,\theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)$$

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

Gradient descent algorithm

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)\right] \rightarrow \frac{\partial}{\partial\theta_0}J(\theta_0,\theta_1)$$

$$\theta_1 := \theta_1 - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)\cdot x^{(i)}\right] \rightarrow \frac{\partial}{\partial\theta_1}J(\theta_0,\theta_1)$$

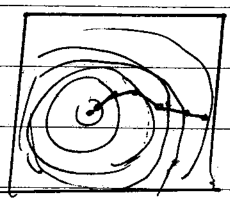
}

update θ_0 and θ_1 simultaneously

"Batch" Gradient Descent

"Batch" : Each step of gradient descent uses all the training examples.

While gradient descent can be susceptible to local minima in general, the optimization problem we have posed here for linear regression has only 1 global and no other local, optima. Thus, gradient descent always converges (assuming α is not too large) to global minimum. Indeed, J is a convex quadratic function. Here is an example of gradient descent as it is run to minimize quadratic function.



The ellipses are the contours of a quadratic function. Also shown is the trajectory taken by gradient descent, which was initialized at (48, 30). "•" in figure (joined by straight line) mark the successive values of θ that gradient descent went through as it converged to its minimum.

No.

Date

Matrices and Vectors

rectangular array of numbers

exp: $\begin{bmatrix} 1402 & 4567 \\ 532 & 924 \\ 122 & 722 \end{bmatrix}$

rows

could ~~be~~ features from learning problem / data

$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$ → 2×3 matrix $\mathbb{R}^{2\times 3}$

Dimension of matrix : num of rows × num of columns

→ 3×2 matrix $\mathbb{R}^{3\times 2}$

Matrix Elements (entries of matrix)

→ numbers inside the matrix

$$A = \begin{bmatrix} 1402 & 191 \\ 1371 & 821 \\ 949 & 1437 \\ 147 & 1448 \end{bmatrix}$$

A_{ij} = "i, j entry" in the i^{th} row, j^{th} column.

A_{11} refer to 1st row 1st column

$A_{11} = 1402$ $\quad$ A_{43} = undefined (error

$A_{12} = 191$

$A_{32} = 1437$

$A_{41} = 147$

No.

Date

vector : An $n \times 1$ matrix (matrix that has only 1 column)

$$y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix} \leftarrow \text{4-dimensioned vector } (\mathbb{R}^4) \quad \underline{n=4}$$

y_i = i^{th} element

$y_1 = 460 \quad y_3 = 315$

1-indexed vs 0-indexed.

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \qquad y = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

Linear Algebra

~~Addition and scalar multiplication~~ (multiply matrix)

Addition

$$\begin{bmatrix} 1 & 0 \\ 2 & 5 \\ 3 & 1 \end{bmatrix} + \begin{bmatrix} 4 & 0.5 \\ 2 & 5 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 5 & 0.5 \\ 4 & 10 \\ 3 & 2 \end{bmatrix}$$

$3 \times 2 \qquad 3 \times 2 \qquad 3 \times 2$ add the matrix of same dimension

$$\begin{bmatrix} 1 & 0 \\ 2 & 5 \\ 3 & 1 \end{bmatrix} + \begin{bmatrix} 4 & 0.5 \\ 2 & 5 \end{bmatrix} = \text{error}$$

$3 \times 2 \qquad 2 \times 2$

Scalar Multiplication (real number)

$$3 \times \begin{bmatrix} 1 & 0 \\ 2 & 5 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 6 & 15 \\ 9 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 4 & 0 \\ 6 & 3 \end{bmatrix} / 4 = \frac{1}{4} \begin{bmatrix} 4 & 0 \\ 6 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{3}{2} & \frac{3}{4} \end{bmatrix}$$

No. ______

Date ______

$$3 \times \begin{bmatrix}1\\4\\2\end{bmatrix} + \begin{bmatrix}0\\0\\5\end{bmatrix} - \begin{bmatrix}3\\0\\2\end{bmatrix} / 3$$

$$= \begin{bmatrix}3\\12\\6\end{bmatrix} + \begin{bmatrix}0\\0\\5\end{bmatrix} - \begin{bmatrix}1\\0\\\frac{2}{3}\end{bmatrix}$$

$$= \begin{bmatrix}2\\12\\\frac{31}{3}\end{bmatrix}$$

$$\begin{bmatrix}4\\6\\7\end{bmatrix} / 2 - 3\begin{bmatrix}2\\1\\0\end{bmatrix}$$

$$= \begin{bmatrix}2\\3\\7/2\end{bmatrix} - \begin{bmatrix}6\\3\\0\end{bmatrix}$$

$$= \begin{bmatrix}-4\\0\\7/2\end{bmatrix}$$ 3-dimensional vector

Matrix vector multiplication

$$\underset{3 \times 2}{\begin{bmatrix}1 & 3\\4 & 0\\2 & 1\end{bmatrix}} \underset{2 \times 1}{\begin{bmatrix}1\\5\end{bmatrix}} = \underset{3 \times 1}{\begin{bmatrix}16\\4\\7\end{bmatrix}} \qquad \begin{bmatrix}1 & 2 & 1 & 5\\0 & 3 & 0 & 4\\-1 & -2 & 0 & 0\end{bmatrix} \begin{bmatrix}1\\3\\2\\1\end{bmatrix} = \begin{bmatrix}14\\13\\-7\end{bmatrix}$$

$1 \times 1 + 3 \times 5 = 16$

$4 \times 1 + 0 \times 5 = 4$

$2 \times 1 + 1 \times 5 = 7$

House sizes:

→ 2104

→ 1416 $\qquad h_\theta(x) = -40 + 0.25x$

→ 1534

→ 852

matrix $\qquad\qquad$ vector $\qquad\qquad\qquad\qquad$ $h_\theta(2104)$

$$\underset{4 \times 2}{\begin{bmatrix}1 & 2104\\1 & 1416\\1 & 1534\\1 & 852\end{bmatrix}} \times \underset{2 \times 1}{\begin{bmatrix}-40\\0.25\end{bmatrix}} = \underset{4 \times 1}{\begin{bmatrix}1 \times -40 + 0.25 \times 2104\\1 \times -40 + 0.25 \times 1416\\1 \times -40 + 0.25 \times 1534\\1 \times -40 + 0.25 \times 852\end{bmatrix}} \leftarrow h_\theta(1416)$$

4 dimensional vector

Octave → prediction = DataMatrix * parameters | ~~for i=1:4~~

No.
Date

matrix-matrix multiplication

$$\begin{bmatrix} 1 & 3 & 2 \\ 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 0 & 1 \\ 5 & 2 \end{bmatrix} = \begin{bmatrix} 11 & 10 \\ 9 & 14 \end{bmatrix}$$

2×3 $\quad 3 \times 2$ $\quad 2 \times 2$

$$\begin{bmatrix} 1 & 3 & 2 \\ 4 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 5 \end{bmatrix} = \begin{bmatrix} 11 \\ 9 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 & 2 \\ 4 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 10 \\ 14 \end{bmatrix}$$

House sizes: Have 3 competing hypotheses:

2104 $\quad$ 1- $h_\theta(x) = -40 + 0.25x$

1416 $\quad$ 2- $h_\theta(x) = 200 + 0.1x$

1534 $\quad$ 3- $h_\theta(x) = -150 + 0.4x$

852

Matrix $\qquad$ Matrix

$$\begin{bmatrix} 1 & 2104 \\ 1 & 1416 \\ 1 & 1534 \\ 1 & 852 \end{bmatrix} \times \begin{bmatrix} -40 & 200 & -150 \\ 0.25 & 0.1 & 0.4 \end{bmatrix} = \begin{bmatrix} 486 & 410 & 692 \\ 314 & 342 & 416 \\ 344 & 353 & 464 \\ 173 & 285 & 191 \end{bmatrix}$$

Matrix multiplication properties

$3 \times 5 = 5 \times 3$ $\rightarrow$ order not important called ~~because~~ "Commutative of multiplication of real numbers"

Let A and B be matrices. in general,

$A \times B \neq B \times A$ (not commutative)

$A \times B$

$m \times n \quad n \times m$

Eg. $\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$

$\begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 2 & 2 \end{bmatrix}$

$A \times B$ is $m \times m$

$B \times A$ is $n \times n$

$3 \times (5 \times 2) = (3 \times 5) \times 2$ "Associative"

Matrix:

$A \times B \times C = \begin{cases} A \times (B \times C) \\ (A \times B) \times C \end{cases}$ same answer (Associative property)

No.

Date

Identity Matrix

1 is identity $1 \times z = z \times 1 = z$

only z

Denoted I (or $I_{n \times n}$)

Examples of identity matrices:

$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ 2×2 $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ 3×3 $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ 4×4 $\Rightarrow$ Informally: $\begin{bmatrix} 1 & & 0 \\ & 1 & \\ 0 & & \ddots \; 1 \end{bmatrix}$

For any matrix A,

$$\underset{m \times n}{A} \cdot \underset{n \times n}{I} = \underset{m \times m}{I} \cdot \underset{m \times n}{A} = \underset{m \times n}{A}$$

Note:

$AB \neq BA$ in general

$AI = IA$ ✓

Inverse and Transpose

I = "identity" $3(3^{-1}) = 1$ $12 \times (12^{-1}) = 1$

Not all numbers have inverse. $\Rightarrow$ $0(\underbrace{0^{-1}}_{\text{undefined}})$

Matrix inverse:

If A is an $m \times m$ matrix, and if it has an inverse,

$A(A^{-1}) = A^{-1}A = I$

Eg. $\begin{bmatrix} 3 & 4 \\ 2 & 16 \end{bmatrix} \begin{bmatrix} 0.4 & -0.1 \\ -0.05 & 0.075 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I_{2 \times 2}$

Matrix transpose

Example: $A = \begin{bmatrix} 1 & 2 & 0 \\ 3 & 5 & 9 \end{bmatrix}$ $A^T = \begin{bmatrix} 1 & 3 \\ 2 & 5 \\ 0 & 9 \end{bmatrix}$

imagine a 45° mirror, and all number reflect here

Let A be an $m \times n$ matrix, $B = A^T$.

B is an $n \times m$ matrix, and

$B_{ij} = A_{ji}$

$B_{12} = A_{21} = 2$

row column

$\begin{bmatrix} 3 & -5 & 4 \end{bmatrix}$ $\begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}$