S-229

Latest Notes

Lecture-2

What is supervised learning problem?

Linear Regression:

Size (feet)	Price ($1000)
2104	400
1416	232
1539	315
852	178

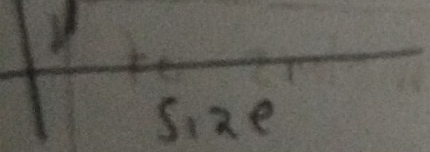


Fig 1

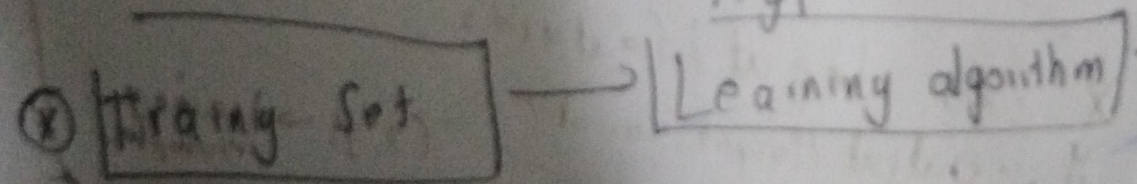


Fig 2

Size → hypothesis → estimated price

How to Represent the Hypothesis

→ $h(x) = \theta_0 + \theta_1 x$ (Linear Regression Hypothesis)

④ → There could be additional features as well

⑧ → $h(x) = \sum_{j=0}^{2} \theta_j x_j$ (Concise form)

where $x_0 = 1$

↳ Here

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} \quad x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$$

Notation:-

→ θ is called the parameters of the learning algorithm

→ m is number of training examples

→ x is input / features

→ y : output / target variables (x, y) → training example

⑨ → $(x^{(i)}, y^{(i)})$ → i^{th} training example

→ $x_j^{(i)}$ where i → training example
j → feature

→ n → # of features (total).

Choose θ s.t $h(x) \approx y$ for training examples

- $h_\theta(x) \rightarrow$ Hypothesis dependents on both x and parameter θ

→ In Linear Regression minimize

$$\min_\theta \left(h_\theta(x) - y \right)^2 \quad \text{Eq.1}$$

$$J(\theta) \rightleftharpoons \min_\theta \frac{1}{2} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2 \quad \text{Eq.1(a)}$$

→ $\min_\theta J(\theta)$.

→ How can we minimize $J(\theta)$?

- Gradient descent - (Baby step to go downhill by taking step steps)

Repeat Until Convergence ←

① Start with some θ (Say $\theta = \vec{0}$).

② Keep changing θ to reduce $J(\theta)$

③ ~~which step~~

③ $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$ (α → Learning rate)

$(j = 0, 1, 2 \cdots n)$.

→ Derivative of function determine the steep step.

↳ Negative sign mean you need to take the step down the hill.

• Assuming one Example:-

$$\rightarrow \frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2} (h_\theta(x) - y)^2$$

$$= 2 \frac{1}{2} (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} (h_\theta(x) - y)$$

$$= (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} (\theta_0 x_0 + \theta x_1 + \cdots \theta_n x_n)$$

$$= (h_\theta(x) - y) \cdot x_j$$

So Now (Gradient descent equation step ③) on Linear Regression

$$\rightarrow \theta_j := \theta_j - \alpha (h_\theta(x) - y) \cdot x_j \quad (Eq. 2)$$

⊙ This is for one example

therefore for m examples:-

$$\rightarrow \theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)} \quad (Eq. 3)$$

→ Now GD will take equation 3 and gradient descent would update θ_j for (0, 1, 2 ... n) until convergence

* The ~~cost~~ $J(\theta)$ is quadratic function therefore it is a big bowl and would have a ~~local~~ global optima.

* If we draw the contour for $J(\theta)$ (SSE)
then it will have prefect ellipsos:

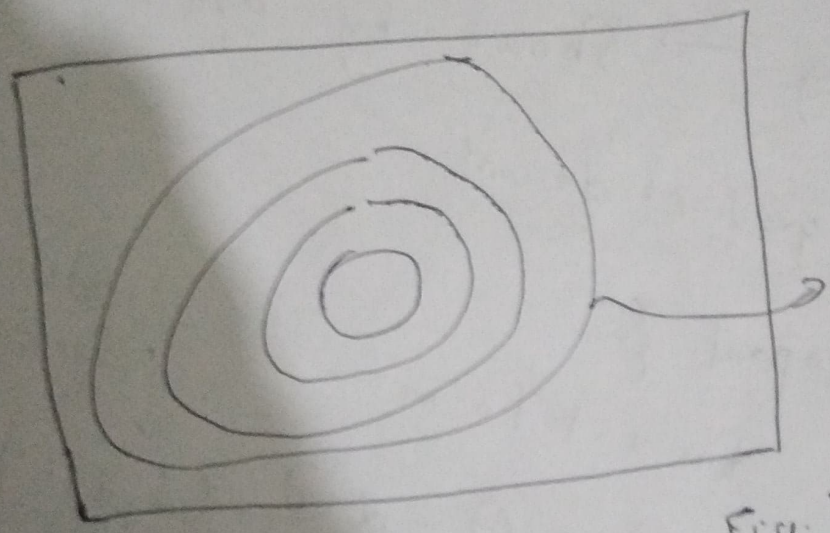


Fig. 3.

* So if you ran gradient descent then the stepdownhill under this cole would always be 90 degree (orthogonal)
↳ Algorithm will converge to global optimia

* $\alpha \rightarrow$ Defines how large will be step.

① Batch gradient descent:

→ Every step of gradient descent will be slow if we take one step of GD on entire data

→ ~~Repeat {~~

→ Shown by equation 3

② Stochastic gradient descent:

⊙ Repeat {

For i=1 to m

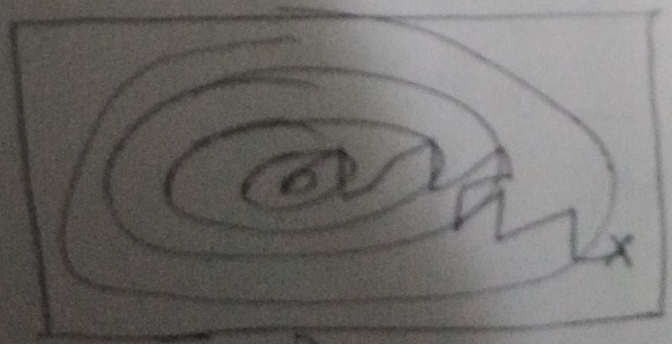
Update for every i

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \cdot x_j^{(i)}$$

} → Eq 4

⊙ Taking gradient descent step just by one example at each time.

③



→ SGD will take a more noisy step than Batch gradient

Fig 4

⊙ Even at global minimum SGD will still oscilates.

⊙ SGD allows you to learn the hypothesis more failed

(*)

(*) Mini-Batch Gradient Descent:-

→ Batch size is of some fix size of b examples

(*) If you have small data set ⟶ Use Batch gradient descent.

(*) Non-iterative Algorithm to solve for parameter θ for linear regression:-

⊙ Normal equation (Goring to show how to derive it)

(*) If $\nabla_\theta J(\theta) = \begin{bmatrix} \frac{\partial J}{\partial \theta_0} \\ \frac{\partial J}{\partial \theta_1} \\ \frac{\partial J}{\partial \theta_2} \end{bmatrix}$

$\theta \in R^{n+1}$

(*) Lets say we have $A \in R^{2\times 2}$

$f(A) = A_{11} + A_{12}^2$ (Eq. 5) $f: R^{2\times 2} \longrightarrow R$

then

$f\left(\begin{bmatrix} 5 & 2 \\ 3 & 9 \end{bmatrix}\right) = 5 + 2^2 = 9$

④ $\nabla_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \cdots & \frac{\partial f}{\partial A_{1n}} \end{bmatrix}$

⑤ $\nabla_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \frac{\partial f}{\partial A_{12}} \\ \frac{\partial f}{\partial A_{21}} & \frac{\partial f}{\partial A_{22}} \end{bmatrix}$ → [illegible]

⑥ We compute derivative of f w.r.t A.

⑦ $\nabla_A f(A) = \begin{bmatrix} 1 & 2A_{12} \\ 0 & 0 \end{bmatrix}$ → taking the previous function f (See Eq-5)

↓

This show the definition of derivative of a matrix with respect to a specific function.

⑧

⑥ How are we going to derive normal equation?

$$\nabla_\theta J(\theta) = \vec{0}$$

⑦ More derivation to help us in above normal equation derivation.

$Tr(A)$ = Sum of diagonal of A element

↓ Trace.

① $tr\ A = tr\ A^T$

② $f(A) = tr\ AB$

$\nabla_A f(A) = B^T$

③ $tr\ AB = tr\ BA$

④ $tr\ ABC = tr\ CAB$.

⑤ $\nabla_A tr\ AA^T C = CA + C^T A$.

⑧ Now returning back to derivation of N.E:

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

$$X = \begin{bmatrix} \text{—} (x^{(1)})^T \text{—} \\ \text{—} x^{(2)T} \text{—} \\ \vdots \\ \text{—} x^{(m)T} \text{—} \end{bmatrix}$$

↕ design matrix

(10)

$$\rightarrow X\theta = \begin{bmatrix} — & x^{(1)T} & — \\ — & x^{(2)T} & — \\ & \vdots & \\ — & x^{(m)T} & — \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}$$

$$= \begin{bmatrix} x^{(1)T}\theta \\ x^{(2)T}\theta \\ \vdots \\ x^{(m)T}\theta \end{bmatrix} = \begin{bmatrix} h_\theta(x^{(1)}) \\ \vdots \\ h_\theta(x^{(m)}) \end{bmatrix}$$

$$\rightarrow y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

→ Turns out:

$$J(\theta) = \frac{1}{2}(X\theta - y)^T(X\theta - y)$$

$$X\theta - y = \begin{bmatrix} h_\theta(x^{(1)}) - y^{(1)} \\ \vdots \\ h_\theta(x^{(m)}) - y^{(m)} \end{bmatrix}$$

remember $z^T z = \sum_i z_i^2$ therefore $(X\theta - y)^T(X\theta - y)$

is sum of square errors.

* Continuing derivation of N.E

$$\nabla_\theta J(\theta) = \nabla_\theta \frac{1}{2}(X\theta - y)^T (X\theta - y)$$

$$= \frac{1}{2} \nabla_\theta (\theta^T X^T - y^T)(X\theta - y)$$

$$= \frac{1}{2} \nabla_\theta (\theta^T X^T X\theta - \theta^T X^T y - y^T X\theta + y^T y)$$

↳ similar to $(ax-b)(ax-b)$
$= a^2x^2 - axb - bax + b^2$

* Taking derivatives w.r.t θ

$$= \frac{1}{2}\left[X^T X\theta + X^T X\theta - X^T y - X^T y \right]$$

Simplify

* $X^T X\theta - X^T y = 0$

* $X^T X\theta = X^T y$ (Normal equation).

$$\theta = (X^T X)^{-1} \cdot X^T y$$ (Optimal value for θ through Normal Equation).

→ if X is invertible then use pseudo inverse.