Model & Cost Function

Training Dataset

↓

Learning Algorithm

↓

input $x^{(i)}$ → h (hypothesis) → output $y^{(i)}$

how to measure hypothesis?

$h(x) = \theta_0 + \theta_1 x$: linear function

linear regression

how to choose the best parameters?

so that $h(x)$ is close to y

均方误差

MSE (mean square error)

$= \frac{1}{2}\sum \rightarrow \sum$ 指平方的均值

$\frac{1}{2}$ 是为了计算梯度时的方便，因为在求导时会和平方项的 $(\ )^2$ 抵消.

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

(training example)

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

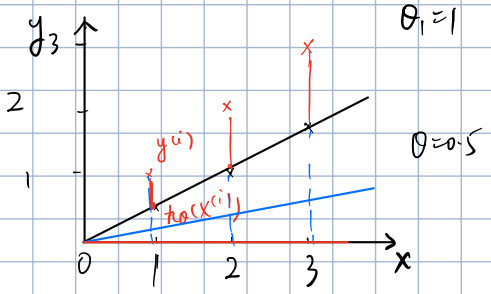
$$\underset{\theta_0,\theta_1}{\text{minimize}}\ J(\theta_0,\theta_1)$$

Cost function

Cost Function

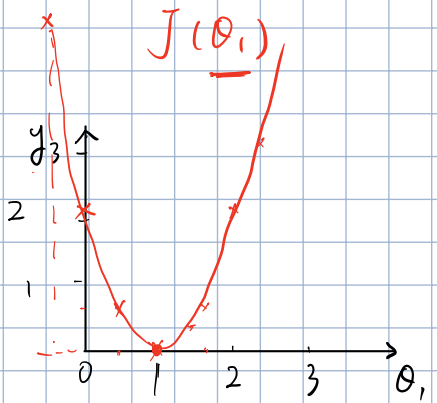
hypothesis function

$h_\theta(x)$



Loss function

$J(\theta_1)$



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

$$= 0$$

$$J(0.5) = \frac{1}{2\times 3}\left(0.5^2 + 1^2 + 1.5^2\right)$$

$$= \frac{3.5}{6} \approx 0.58$$

$$J(0) = \frac{1}{2\times 3}(1+4+9) = \frac{14}{6} \approx 2.3$$

cost function

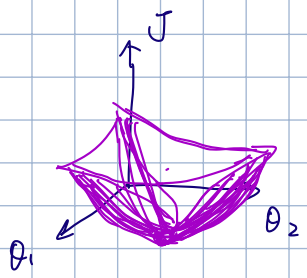
minimize $J(\theta_1)$, will find the best parameter.

Hypothesie: $H(x) = \theta_0 + \theta_1 x$

Parameter: θ_0, θ_1

Loss Function: $J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}[h(x^{(i)}) - y^{(i)}]^2$

Goal: Minimize $J(\theta_0, \theta_1)$ over θ_0, θ_1

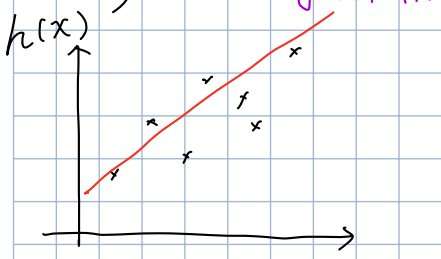


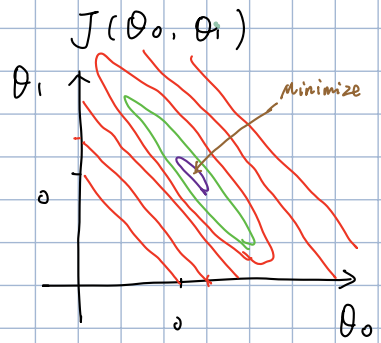
用等高线 (contour plot)

在2D上表示3D图象

同条线上, $J(\theta_0, \theta_1)$ 即Loss相同

2个参数:





Parameter Learning

gradient descent Algorithm: ← use to estimate the parameters in the hypothesis function

Any function → $J(\theta_0, \theta_1)$...

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

↓

repeat until convergence

Simultaneously update:

1. 参数同步更新, 而非更新一个后, 用该项去更新另一个.
2. 对初始位置敏感, 可能会收敛到局部最小 local minimum

(步长会自动越来越短, 即越来越精细).

no need to decrease α over time,

→ because gradient descent will automatically take smaller steps as the derivative get smaller.

参数: α 过小: 下降速度太慢

α 过大: 可能导致不收敛(发散) fail to converge

$\frac{d}{\theta_1} J(\theta_1)$ approaches 0 as we approach the bottom of our convex function.

So that $\theta_1 := \theta_1 - \alpha * 0$

Gradient descent & Cost Function

$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$

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

$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}[h(x^{(i)}) - y^{(i)}]^2$

$= \frac{1}{2m}\sum_{i=1}^{m}[\theta_0 + \theta_1 x^{(i)} - y^{(i)}]^2$

$j=0$: $\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}[h(x^{(i)}) - y^{(i)}]$

$j=1$: $\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$

$= \frac{1}{m}\sum_{i=1}^{m}[h(x^{(i)}) - y^{(i)}] \cdot x^{(i)}$

→ Gradient Descent Algorithm:

repeat until convergence {

$\theta_0 := \theta_0 - \alpha \cdot \frac{1}{m}\sum_{i=1}^{m}[h_\theta(x^{(i)}) - y^{(i)}]$ ← $\frac{\partial}{\partial\theta_0} J(\theta_0,\theta_1)$ update θ_0 and θ_1 simultaneously.

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

}

[即计算完一个参数后，不要代回原来式子，更新参数都是用旧输入]

in linear regression:

Cost function are convex function so that it doesn't have any local optimal except for the one global optimum.

实际中，有时找到局部最小也够用了。

"Batch" Gradient Descent:

each step of gradient descent using all the training samples.

Linear Algebra Review:

对于 $h(x) = \theta_0 + \theta_1 x$.

Prediction = Datamatrix * Parameters.

x (房间面积) $= \begin{pmatrix}100\\121\\195\\130\end{pmatrix}$, $\begin{bmatrix}\theta_0\\\theta_1\end{bmatrix}$

$\begin{pmatrix}1 & 100\\1 & 121\\1 & 195\\1 & 130\end{pmatrix} \cdot \begin{pmatrix}50\\0.25\end{pmatrix} = \begin{pmatrix}1\times50 + 100\times0.25\\1\times50 + 121\times0.25\\\vdots\end{pmatrix}$

补充 ← 4×2 · 2×1 → 4×1

矩阵向量乘法: □ × □ = □

5×2 · 2×1 → 5×1

"相乘"

两层 决定能否相乘; 结层定型.

矩阵矩阵乘法，就像矩阵向量乘法的并行式操作: □ × □ = □

5×2 · 2×3 → 5×3

举例: 对于 $h_1(x) = 50 + 0.25x$, $h_2(x) = 70 + 0.3x$, $h_3(x) = 20 + 0.7x$

$\begin{pmatrix}1 & 100\\1 & 121\\1 & 195\\1 & 130\end{pmatrix} \cdot \begin{pmatrix}50 & 70 & 20\\0.25 & 0.3 & 0.7\end{pmatrix} =$

补充 ← 2×3

the result of the prediction of the first hypothesis

the third hypothesis.

4×2

↑ 4×3

the second hypothesis

Identity Matrix:

$A \cdot I = I \cdot A = A$ → 其实是2个不一样的矩阵.

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

$AB \neq BA$ in general except A/B is I.

$(A \times B) \times C = A \times (B \times C)$ ✓

Matrix Inverse.

eg: $12 \times \frac{1}{12} = 1$; $0 \times X = 1$?

$\rightarrow = 12^{-1}$

$A \cdot A^{-1} = A^{-1} A = I$, only if A is a square matrix ($m \times m$). 零矩阵不存在逆矩阵.

Matrix Transpose

$A = \begin{pmatrix} 1 & 2 & 0 \\ 3 & 5 & 9 \end{pmatrix}$ $A^T = \begin{pmatrix} 1 & 3 \\ 2 & 5 \\ 0 & 9 \end{pmatrix}$ $A_{ij} = B_{ji}$

$= A$ 2×3 $= B$, 3×2