ML OL Course 1

introduction

Linear Regression with single variable.

What is Machine learning?

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

Tom miechell

Supervised learning: Regression:

Classification: Discrete Valued output (0 or 1)

- Define supervised learning as problems where the desired output is provided for examples in the training set.
- Define regression as a subset of supervised learning problems where the output is continuous
- Define classification as a subset of supervised learning problems where the output is discrete.

Octave?

Unsupervised learning:

月想衣裳花想容.

路人纷纷而匆匆

抬首 ~~抚胸顿行~~ 苦愁容.

何天长叹 世道艰 / 真苍茫.

繁星吹落如雨

烟花飘散如雾

古道浮躁几许

斜阳两侧蹊径

流水岂知落花意/情里

眼眸

Machine learning — E: experience
T: Task
P: performance

A computer program is said ~~that~~ 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 measured by P, improves with experience E.

Supervised learning: desirered output < continuous regression / discrete classification

Unsupervised learning: X desired output

linear regression: univariate

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

Cost Function $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$

Gradien Decent

m = Number of training examples

X Features

Y output

(x, y) – ~~one~~ training examples

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

Training Set

↓

learning algorithm

↓

size of house → $\boxed{h}$ → Estimated price

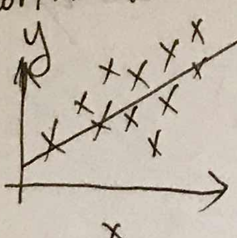
x hypothesis (estimated value of y)

h maps from x's to y's

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

Shorthand : $h(x)$

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



linear regression with one variable (x)

univariate linear regression

↖ fancy name : One variable

Cost function

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

θ_i's : Parameters.

how to choose θ_0 and θ_1

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

# of training examples

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

training example

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

cost function

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

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

↑ square error function

Countour Plots / Figures

Gradien decent

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 hopefully end up at a minimum

---

Gradient decent Algorith

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

learning rate

Assignment	Truth Assertion
$a := b$	$a = b$
$a := a+1$	$a = a+1$

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

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

$\theta_0 := temp0$

$\theta_1 := temp1$

Correct: simultaneous update

Gradient decent can converge to a local minimum, even with the learning rate α fixed.