Week01

Stanford机器学习公开课

---

1. Introduction

1) What is Machine Learning?

Arthur Samuel described it as: "the field of study that gives computers the ability to learn without being explicitly programmed." This is an older, informal definition.

Tom Mitchell provides a more modern definition: "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 measured by P, improves with experience E."

2) Types of Problems

Supervised Learning -- Labeled Data

- Regression: input -> continuous output
- Classification: input -> discrete categories

Unsupervised Learning -- Unlabeled Data

- Clustering
- Others (Associative memory...)

2. Linear Regression with One Variable

1) Model Representation

Suppose we have a training set, say 47 records of house information, shown in the table below, each of which contains the house price and the area of the house. We want to predict the house price for any given area.

line number	Size(x)	Price in 1000's(y)
1	2104	460
2	1416	232
3	1534	315
...	...	...

Some Notation:
m - number of training examples, in our case, $m = 47$.
(x, y) - a single training example
$(x^{(i)}, y^{(i)})$ - the i^{th} training example (in our case, $x^{(1)} = 2104$ and $y^{(1)} = 460$)

By the way, we start counting from 1, just like the convention in Matlab/Octave we use.

After careful consideration, we decide to use a straight line to fit the data. That is, we make a hypothesis that $h_\theta(x) = \theta_0 + \theta_1 x$.

We use $h_\theta(x)$ and its short-handed form $h(x)$ interchangably.

In regression problems, **hypothesis** is a function takes the input and output the estimated value, or **mapping** from input to output.

This model is called **linear regression with one variable**, or **univariate linear regression**.

2) Cost Function

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

So, we want to solve the following minimization problem:

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

where

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

And this is **squared error cost function**, which is the most common one in both machine learning and statistics.

Define

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

, so the optimization problem becomes

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

.

A quick summary:

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

Parameters:
θ_0, θ_1

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

Goal:
$\min_{\theta_0,\theta_1} J(\theta_0, \theta_1)$

3. Gradient Descent

Gradient descent is an algorithm that minimizes the cost function.

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

2) Gradient descent algorithm

repeat until convergence {

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

}

When we say update, we mean **simultaneous update**.

3) Gradient Descent for Linear Regression

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (\theta_0 + \theta_1 x^{(i)} - y^{(i)})$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (\theta_0 + \theta_1 x^{(i)} - y^{(i)}) x^{(i)}$$

}