

# Machine Learning - Week 2

赵燕

## 目录

<b>1</b>	<b>Multivariate Linear Regression</b>	<b>2</b>
1.1	Multiple Features . . . . .	2
1.2	Gradient Descent for Multiple Variables . . . . .	3
1.3	Gradient Descent in Practice I-Feature Scaling . . . . .	4
1.4	Gradient Descent in Practice II-Learning Rate . . . . .	5
1.5	Features and Polynomial Regression . . . . .	5
<b>2</b>	<b>Computer Parameters Analytically</b>	<b>5</b>
2.1	Normal Equation . . . . .	5
2.2	Normal Equation Noninvertibility . . . . .	5
<b>3</b>	<b>Submitting Programming Assignments</b>	<b>5</b>
3.1	Working on and Submitting Programming . . . . .	5
3.2	Programming tips from Mentors . . . . .	5

# 1 Multivariate Linear Regression

## 1.1 Multiple Features

根据之前的例子，我们对房价模型增加更多的特征，例如房间数楼层等，构成了一个含有多个变量的模型，模型中的特征值为  $(x_1, x_2, \dots, x_n)$ ：

**Multiple features (variables).**

Size (feet <sup>2</sup> ) $x_1$	Number of bedrooms $x_2$	Number of floors $x_3$	Age of home (years) $x_4$	Price (\$1000) $y$
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
...	...	...	...	...

Notation:

- $\rightarrow n$  = number of features  $n = 4$
- $\rightarrow x^{(i)}$  = input (features) of  $i^{th}$  training example.
- $\rightarrow x_j^{(i)}$  = value of feature  $j$  in  $i^{th}$  training example.

Handwritten notes:

- $m = 47$  (number of training examples)
- $x^{(2)} = \begin{bmatrix} 1416 \\ 3 \\ 2 \\ 40 \end{bmatrix}$  (feature vector for the 2nd example)
- $x_3^{(2)} = 2$  (value of the 3rd feature for the 2nd example)

图 1: 多变量房价模型

注释:

(1)  $n$ : 特征的数量

(2)  $x^{(i)}$ : 第 $i$ 个训练样本，是特征矩阵中的第 $i$ 行，是一个向量 (vector)；例如：上图中的

$$x^{(2)} = \begin{bmatrix} 1416 \\ 3 \\ 2 \\ 40 \end{bmatrix} \quad (1)$$

(3)  $x_j^{(i)}$ : 特征矩阵中第 $i$ 行的第 $j$ 个特征值，也就是第 $i$ 个训练样本实例的第 $j$ 个特征，如上图的 $x_3^{(2)} = 2$ 。

多变量的假设函数:

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \quad (2)$$

这个公式中有你 $n+1$ 个参数和 $n$ 个变量，为了能使公式简化，引入 $x_0=1$ ，则公式转化为:

$$h_{\theta}(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \quad (3)$$

此时模型中的参数是一个n+1维的变量，任何一个训练实例都是n+1维的向量，特征矩阵X的维度是m\*(n+1)。公式可以简化为：

$$h_{\theta}(x) = [\theta_0 \quad \theta_1 \quad \dots \quad \theta_n] \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \theta^T X \quad (4)$$

$$h_{\theta}(x) = \theta^T X \quad (5)$$

→  $h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$   
 For convenience of notation, define  $x_0 = 1$ . ( $x_0^{(i)} = 1$ )

$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1}$       $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$

$h_{\theta}(x) = \theta_0 x_0 + \theta_1 x_1 + \dots + \theta_n x_n$   
 $= \boxed{\theta^T x}$

$\begin{bmatrix} \theta_0 & \theta_1 & \dots & \theta_n \end{bmatrix}$  (1×(n+1) matrix)  $\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$  ((n+1)×1 vector)

图 2: 多变量假设函数

## 1.2 Gradient Descent for Multiple Variables

与单变量线性回归类似，在多变量线性回归中也构建一个代价函数，则这个代价函数是所有建模误差的平方和，即：

$$J(\theta_1, \theta_2, \dots, \theta_n) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \quad (6)$$

其中：

$$h_{\theta}(x) = \theta^T X = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \quad (7)$$

多变量梯度算法定义：n=1和n<sub>i</sub>=1的情况其实是一样的

repeat until convergence: {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_0^{(i)} \quad (8)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_1^{(i)} \quad (9)$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_2^{(i)} \quad (10)$$

.....

}

In others words:

repeat until convergence: {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad (11)$$

$$\text{for } j := 0, 1, 2, \dots, n \quad (12)$$

}

单变量与多变量的比较: The following image compares gradient descent with one variable to gradient descent with multiple variables:

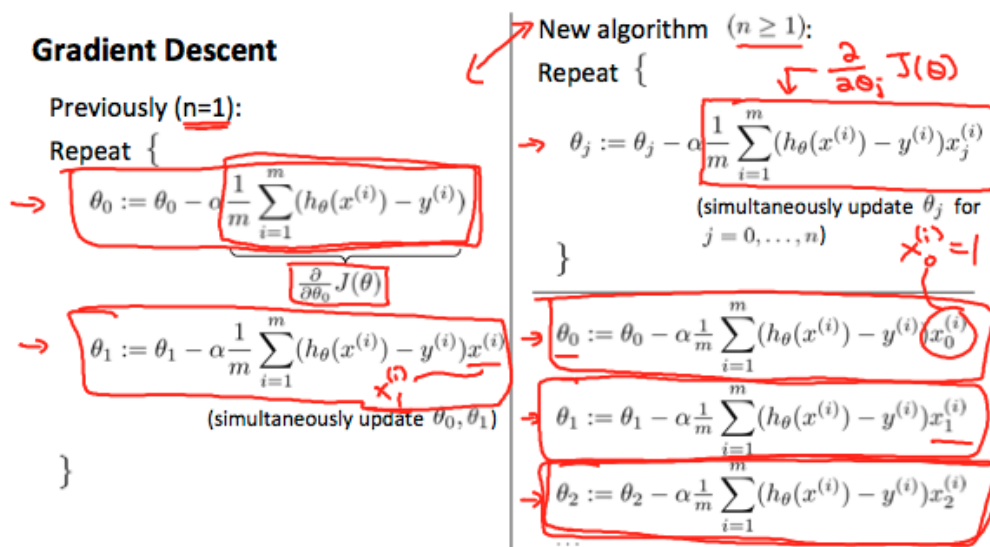


图 3: 单变量与多变量的梯度下降算法对比

### 1.3 Gradient Descent in Practice I-Feature Scaling

特征缩放可以将梯度下降的速度加快，将梯度下降收敛所需的循环次数减少。

以房价问题为例，假设我们使用两个特征,房屋的尺寸和房间的数量,尺寸的值 0-2000 平方英尺,而房间数量的值则是 0-5,以两个参数分别为横纵坐标,绘制代价函数的等高线图，如图所示，代价函数的等高线图很扁，又细又长，梯度下降算法需要非常多次的迭代才能收敛。

解决的额方法是尝试将所有特征的尺度都尽量缩放到-1到1之间。

## Feature Scaling

Idea: Make sure features are on a similar scale.

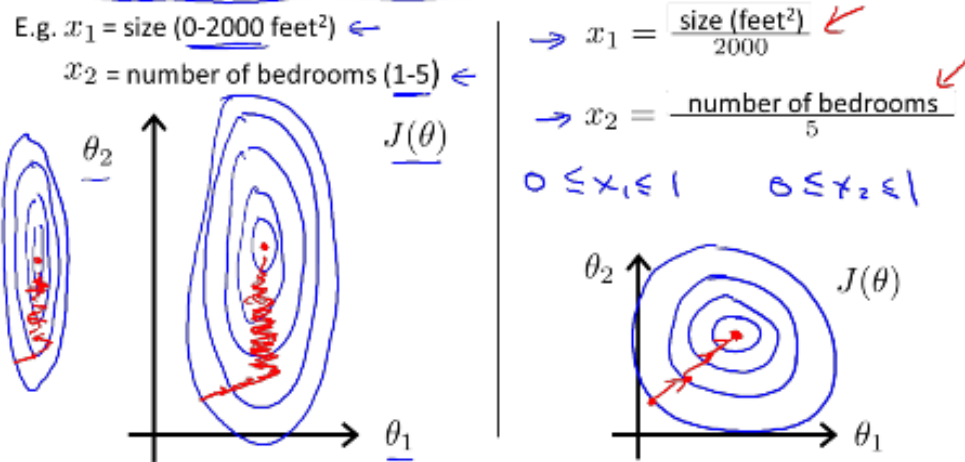


图 4: 多变量假设函数

Two techniques to help with this are feature scaling and mean normalization. Feature scaling involves dividing the input values by the range (i.e. the maximum value minus the minimum value) of the input variable, resulting in a new range of just 1. Mean normalization involves subtracting the average value for an input variable from the values for that input variable resulting in a new average value for the input variable of just zero. To implement both of these techniques, adjust your input values as shown in this formula:

$$x_i = \frac{x_i - \mu_i}{s_i} \quad (13)$$

$\mu_i$  :is the average of all the values for feature (i);

$s_i$  :is the range of values (max - min), or  $s_i$  is the standard deviation(标准差)

Note that dividing by the range, or dividing by the standard deviation, give different results. The quizzes in this course use range - the programming exercises use standard deviation.

For example: if  $x_i$  represents housing prices with a range of 100 to 2000 and a mean value of 1000, then:

$$x_i = \frac{\text{price} - 1000}{900} \quad (14)$$

### 1.4 Gradient Descent in Practice II-Learning Rate

### 1.5 Features and Polynomial Regression

## 2 Computer Parameters Analytically

### 2.1 Normal Equation

### 2.2 Normal Equation Noninvertibility

## 3 Submitting Programming Assignments

### 3.1 Working on and Submitting Programming

### 3.2 Programming tips from Mentors