# Note: Linear Regression with Multiple Variables

Sun Zhao

January 10, 2013

# 1 Multiple Features

Univariate linear regression takes only one variable to predict values. Mostly, values are relevant to multiple features other than one feature. Multiple features are powerful than single feature to predict values. An example of multiple features is shown in Table 1. A house's price is now related to the size, number of bedrooms, number of floors, and age of it. Each row is an training example with left three feature columns and right one value column. We use n to denote the number of features, $x^{(i)}$ to denote the features of the $i^{th}$ training example and $x_j^{(i)}$ to denote the value of feature j in $i^{th}$ training example. For example, $x^{(2)}$ equals $[1416 \quad 3 \quad 2 \quad 40]^T$ and $x_3^{(2)}$ equals 2. Since the hypothesis of univariate linear regression is $h_\Theta(x) = \Theta_0 + \Theta_1 x$, that of multivariate linear regression becomes (1).

Table 1:

| size($feet^2$) | # of bedrooms | # of floors | Age(years) | Price($1000) |
|---|---|---|---|---|
| 2014 | 5 | 1 | 45 | 460 |
| 1416 | 3 | 2 | 40 | 232 |
| 1534 | 3 | 2 | 30 | 315 |
| 852 | 2 | 1 | 36 | 178 |
| ... | ... | ... | ... | ... |

$$h_\Theta(x) = \Theta_0 + \Theta_1 x_1 + \Theta_2 x_2 + \ldots + \Theta_n x_n \tag{1}$$

Let $x_0^{(i)} = 1$, $\Theta = [\Theta_0 \quad \Theta_1 \ldots \Theta_n]^T$, and $x = [x_0 \quad x_1 \quad \ldots x_n]^T$, then (1) is refined as (2).

$$h_\Theta(x) = \Theta^T x \tag{2}$$

# 2 Gradient Descent

Refer to (1) we can infer the cost function of multivariate linear regression shown in (3).

$$J(\Theta_0, \Theta_1, \ldots, \Theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\Theta(x^{(i)}) - y^{(i)})^2 \tag{3}$$

And the gradient descent algorithm is adapted to:

Algorithm1: Gradient Descent for Multivariate Linear Regression

Repeat{

$\Theta_j = \Theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\Theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)}$

}

# 3 Practical Issues

## 3.1 Feature Scaling

The idea of feature scaling is to make sure that all features are on a similar scale. If one feature's range is significantly dominating the others', the regression process will converge at a very slow speed. Generally, making the scale of all features is around [-1, 1] is a good choice. Methods for normalization includes dividing values by its ranges or first decreasing by its mean values and then dividing by its range. In a word, we can normalize x from [min, max] with x/(max-min) or (x-average([min, max]))/(max-min).

## 3.2 Learning Rate

The core function of gradient descent is $\Theta_j = \Theta_j - \alpha \frac{\partial}{\partial \Theta_j} J(\Theta)$. When running gradient descent algorithm, we need to choose an appropriate $\alpha$ and make sure the algorithm is working correctly. Teacher recommended to plot the value of $J(\Theta)$ against the number of regression iterations. If the plotted curve decreases faster as Fig. 1(A), then we can confirm $\alpha$ and the algorithm all works well. If the plotted curve decreases slow as Fig. 1(B), we need to increase the value of $\alpha$ a little more. Otherwise, the plotted curve increase as Fig. 1(C) or vibrates as Fig. 1(D), we have to decrease the value of $\alpha$.

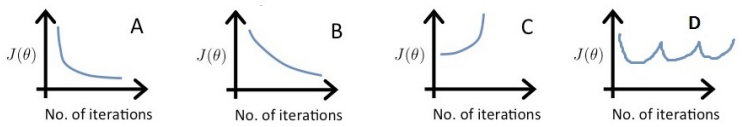


Figure 1:

## 3.3 Choosing Features

Generally, we can adopt three rules to choose features:

- Feature generalization.
- Feature reduction.
- Polynomial feature

Feature generalization suggests that generalized feature is often better than multiple combined features. For example, when predicting the price of a house, the area is a good generalized feature than combined features with frontage and depth. When talking about feature reduction, we mean that features relevant to others should be removed and treated as redundant features. Since values may not be linearly relevant to features, we can use polynomial features such

as square features, cubic features and so on. As polynomial feature is adopted, the hypothesis function may be like $h_\Theta(x) = \Theta_0 + \Theta_1 x_1 + \Theta_2 x_2^2 + \Theta_3 x_3^3$. Let $y_1 = x_1, y_2 = x_2^2, y_3 = x_3^3$, the hypothesis is reduced to $h_\Theta(y) = \Theta_0 + \Theta_1 y_1 + \Theta_2 y_2 + \Theta_3 y_3$ which is a linear regression hypothesis.

# 4 Normal Equation

Gradient descent solves linear regression literately, however, normal equation solves it in a analytical way. Recall that the cost function of linear regression is $J(\Theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\Theta(x^{(i)}) - y^{(i)})^2$ and according to normal equation, let $\frac{\partial}{\partial\Theta_j}J(\theta_j, \Theta_1) = 0$ (for j = 0 ... m), and we will get a system of equations. Applying $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\Theta(x^{(i)}) - y^{(i)})^2$, the linear algebra form of normal equation is:

$$X^T X\Theta - X^T Y = 0 \tag{4}$$

thus,

$$\Theta = (X^T X)^{-1} X^T Y \tag{5}$$

The advantages and disadvantages comparison between gradient descent and normal equation is shown in Table2.

Table 2:

| Gradient Descent | Normal Equation |
|---|---|
| Need to choose $\alpha$ | No need to choose $\alpha$ |
| Need many iterations | No iterations |
| Works well when n is large | Slow if n is large |
| No need to compute matrix | Need to compute $(X^T X)^{-1}$ |

# 5 Summary

Multivariate linear regression takes advantage of multiple features and is much powerful than univariate linear regression. When using gradient descent algorithms, we should take care of choosing appropriate features and learning rate. If the number of examples is not very large, normal equation is a faster alternative.