§ Machine Learning — Week 2 Assignment

1. computeCost.m

2.2.1 Update Equations

The objective of linear regression is to minimize the cost function

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

where the hypothesis $h_\theta(x)$ is given by the linear model

$$h_\theta(x) = \theta^T x = \theta_0 + \theta_1 x_1$$

```
function J = computeCost(X, y, theta)
%COMPUTECOST Compute cost for linear regression
%   J = COMPUTECOST(X, y, theta) computes the cost of using theta as the
%   parameter for linear regression to fit the data points in X and y

% Initialize some useful values
m = length(y); % number of training examples

% You need to return the following variables correctly
J = 0;

% ====================== YOUR CODE HERE
======================
% Instructions: Compute the cost of a particular choice of theta
%               You should set J to the cost.

J=1/(2*m)*((X*theta-y)'*(X*theta-y));
%
=========================================================================
=================

end
```

$$\underset{97\times2}{X} = \begin{bmatrix} 1 & 6.1101 \\ 1 & 5.3277 \\ \vdots & \vdots \\ 1 & 5.4369 \end{bmatrix} \quad \underset{2\times1}{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \rightarrow \underset{97\times1}{X*\theta} = \begin{bmatrix} \theta_1 + 6.1101\theta_2 \\ \theta_1 + 5.3277\theta_2 \\ \vdots \\ \theta_1 + 5.4369\theta_2 \end{bmatrix}$$

$$\underset{97\times1}{Z} = X*\theta - y = \begin{bmatrix} \theta_1 + 6.1101\theta_2 - 17.592 \\ \vdots \\ \theta_1 + 5.4389\theta_2 - 0.61705 \end{bmatrix} \rightarrow h_\theta(x^{(i)}) - y^{(i)}$$

$Z' \to 1\times 97$

$$\Rightarrow \underset{1\times 1}{Z'*Z} = (X*\theta-y)'*(X*\theta-y) = \sum_{i=0}^{m}\left(h_\theta(X^{(i)})-y^{(i)}\right)^2$$

2. gradient Descent.m

$$\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})x_j^{(i)} \qquad \text{(simultaneously update } \theta_j \text{ for all } j\text{)}.$$

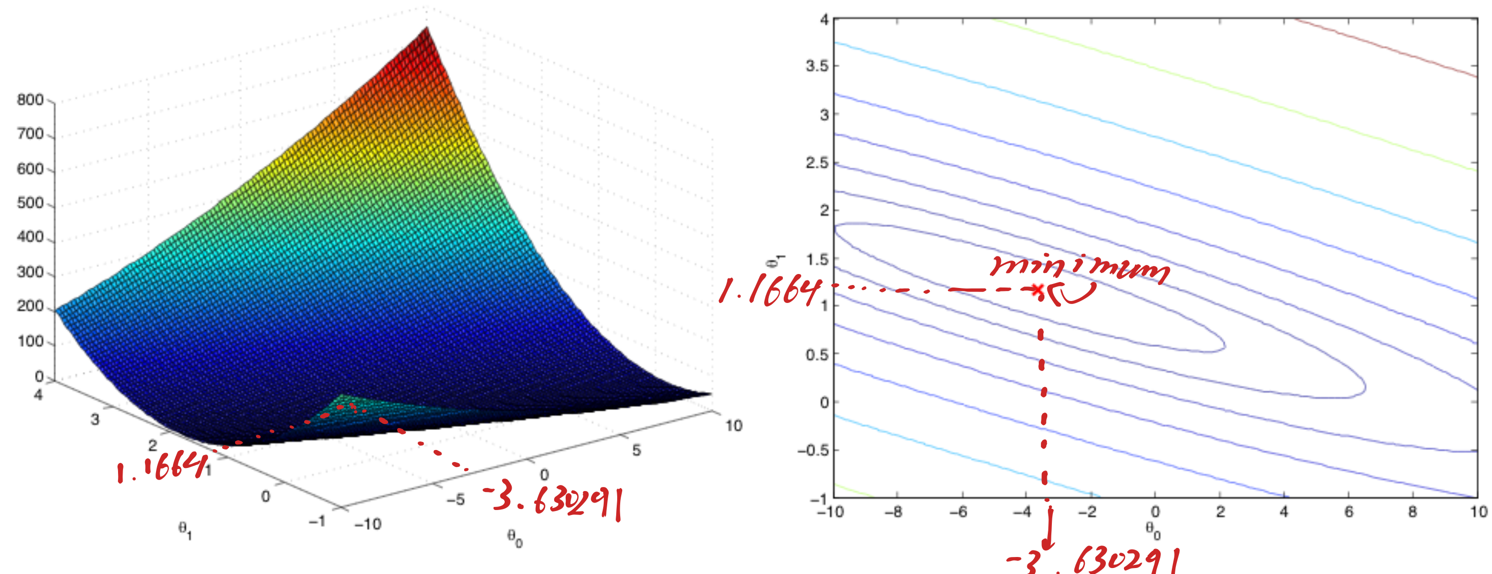
```
function [theta, J_history] = gradientDescent(X, y, theta, alpha, num_iters)
%GRADIENTDESCENT Performs gradient descent to learn theta
%   theta = GRADIENTDESCENT(X, y, theta, alpha, num_iters) updates theta by
%   taking num_iters gradient steps with learning rate alpha

% Initialize some useful values
m = length(y); % number of training examples
J_history = zeros(num_iters, 1);

for iter = 1:num_iters

    % ====================== YOUR CODE HERE
======================
    % Instructions: Perform a single gradient step on the parameter vector
    %               theta.
    %
    % Hint: While debugging, it can be useful to print out the values
    %       of the cost function (computeCost) and gradient here.
    %
    theta = theta-alpha/m*X'*(X*theta-y);
    %
============================================================
===

    % Save the cost J in every iteration
    J_history(iter) = computeCost(X, y, theta);

end

end
```

$$\underset{2\times 1}{\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}} - \frac{\alpha}{m}\underset{2\times 97}{\begin{bmatrix}1 & 1 & \cdots & 1\\ 6.1101 & 5.5277 & \cdots & 5.4369\end{bmatrix}}\cdot\underset{97\times 1}{\begin{bmatrix}\theta_1+6.1101\theta_2-17.592\\ \vdots\\ \theta_1+5.4369\theta_2-0.61705\end{bmatrix}}$$

$$= \Big[\theta_1 - \frac{\alpha}{m}\sum^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)$$

$$\left[\theta_2 - \frac{\alpha}{m}\sum_{i=1}^{m}\left(h_\theta(X^{(i)}) - y^{(i)}\right)X_1^{(i)} \right]_{2\times 1}$$

3. visualizing $J(\theta)$



(a) Surface (b) Contour, showing minimum

In 3D surface, θ_0 and θ_1 can be changed from different value, which reduce the value of $J(\theta)$, and finally $J(\theta)=0$

4. optional exercises

not different, pay attention that gradient descent needs feature scaling while normal equation doesn't need it

$x_i := \frac{x_i - \mu_i}{s_i}$ average

standard deviation

(1) gradient descent:

$$J(\theta) = \frac{1}{2m}\left(X\theta - \vec{y}\right)^T\left(X\theta - \vec{y}\right)$$

```
J=1/(2*m)*((X*theta-y)'*(X*theta-y));
```

(2) normal equation:

$$\theta = \left(X^TX\right)^{-1}X^T\vec{y}.$$

```
theta = pinv(X'*X)*X'*y;
```