Machine Learning and Data Analytics
ME 5013- Fall 2019

Lecture 06

- Gradient Descent cont.
- Polynomial Regression

The University of Texas at San Antonio™

Adel Alaeddini, PhD
Associate Professor of Mechanical Engineering
Advanced Data Engineering Lab
adel.alaeddini@utsa.edu

Disclosure: some of the slides are adopted from Dr. Andrew Ng lecture notes on "Machine Learning" at Stanford university

Gradient descent for multiple regression

Hypothesis: $h_\beta(x) = \beta_0 + \beta_1 x$

Cost Function: $\frac{1}{2n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x^{(i)})^2$

x_i

Parameters: β_0, β_1

Gradient descent:

Repeat until convergence{

$$\beta_0 := \beta_0 + \alpha \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x^{(i)})$$

$$\beta_1 := \beta_1 + \alpha \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x^{(i)})\, x^{(i)}$$

}

Hypothesis: $h_\beta(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$

Cost Function: $\frac{1}{2n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_1^{(i)} - \cdots - \beta_p x_p^{(i)})^2$

Parameters: $\beta_0, \beta_1, \ldots, \beta_p$

Gradient descent:

Repeat until convergence{

$$\beta_0 := \beta_0 + \alpha \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_1^{(i)} - \cdots - \beta_p x_p^{(i)})$$

$$\beta_1 := \beta_1 + \alpha \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_1^{(i)} - \cdots - \beta_p x_p^{(i)})\, x_1^{(i)}$$

…

$$\beta_p := \beta_p + \alpha \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_1^{(i)} - \cdots - \beta_p x_p^{(i)})\, x_p^{(i)}$$

}

Feature Scaling

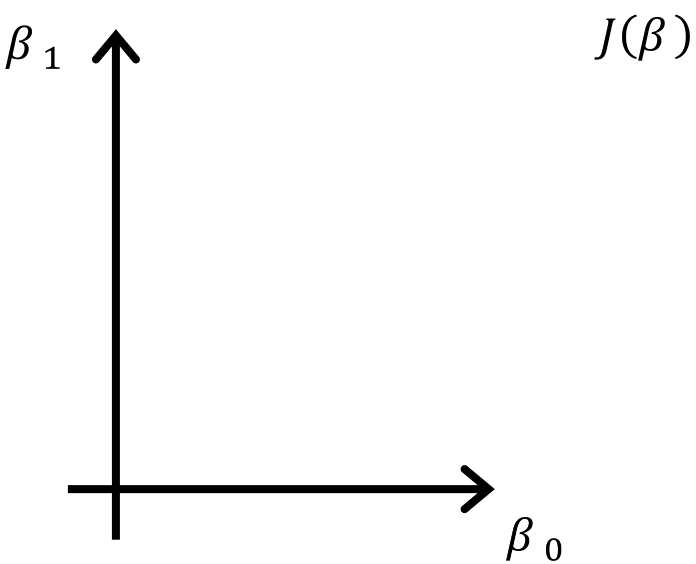
Idea: Make sure features are on a similar scale.

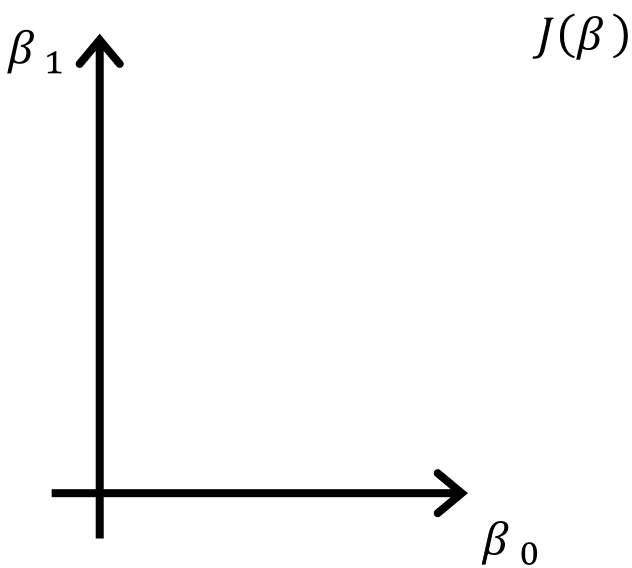
E.g. x_1 = size (0-2000 feet^2)

x_2 = number of bedrooms (1-5)

$$x_1 = \frac{\text{size (feet}^2\text{)}}{2000}$$

$$x_2 = \frac{\text{number of bedrooms}}{5}$$





Feature Scaling

Get every feature into approximately a $-1 \le x_i \le 1$ range.

Mean normalization

- Replace x_i with x_i-$\overline{x_i}$, where $\overline{x_i}$ is the average value of x_i in the training set, to make features have approximately zero mean (Do not apply to x_0).
- Divide x_i by R_i, where R_i is the range of x_i in the training set,

  OR

  Divide x_i by S_i, where S_i is the standard deviation of x_i in the training set

E.g. $x_2 = \frac{\#bedrooms - 2}{5}$

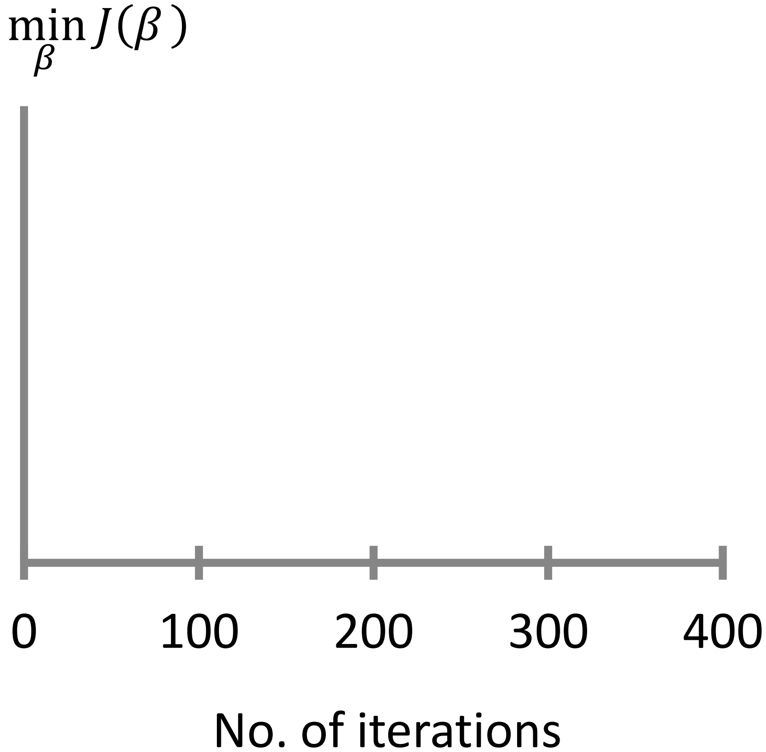
Gradient descent

$$\beta_j := \beta_j - \alpha \frac{\partial}{\partial \beta_j} J(\beta)$$

- "Debugging": How to make sure gradient descent is working correctly.
- How to choose learning rate α.

Gradient descent in practice II: Learning rate

Making sure gradient descent is working correctly.

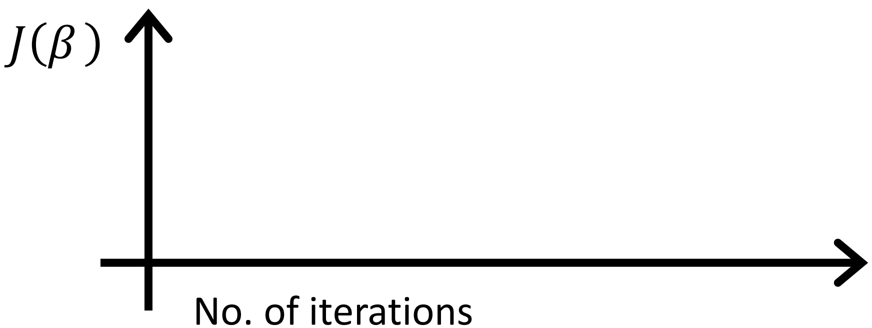


Example automatic convergence test:

Declare convergence if $J(\beta)$ decreases by less than 10^{-3} in one iteration.

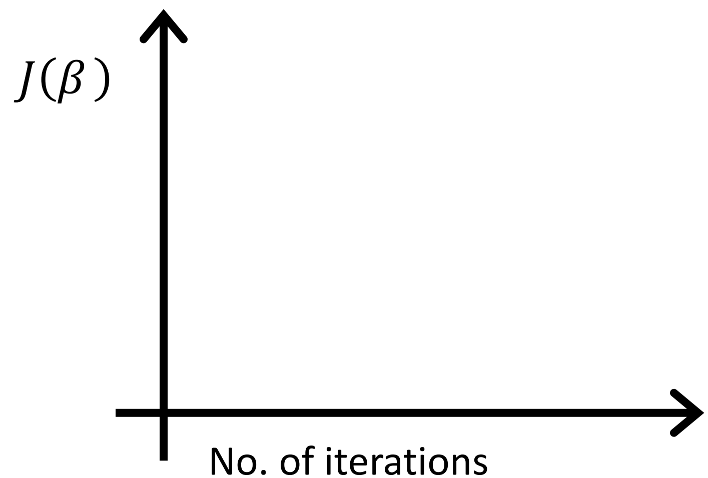
Gradient descent in practice II: Learning rate

Making sure gradient descent is working correctly.



Gradient descent not working.

Use smaller α



- For sufficiently small α, $J(\beta)$ should decrease on every iteration.
- But if α is too small, gradient descent can be slow to converge.

Gradient descent in practice II: Learning rate

Summary:

- If α is too small: slow convergence.
- If α is too large: $J(\beta)$ may not decrease on every iteration; may not converge.

To choose α, try

$$\ldots, 0.001, \qquad , 0.01, \qquad , 0.1, \qquad , 1, \ldots$$

Gradient descent vs normal equation

n **training examples,** p **features.**

Gradient Descent

- Need to choose α.
- Needs many iterations.
- Works well even when p is large.
- More robust to linearly dependencies

Normal Equation

- No need to choose α.
- Don't need to iterate.
- Need to compute $(X^T X)^{-1}$
- Slow if p is very large.
- Less robust to linearly dependencies

Gradient descent vs normal equation

What if $X^T X$ is non-invertible?

- Redundant features (linearly dependent).
  E.g. $x_1 =$ size in feet^2
  $x_2 =$ size in m^2

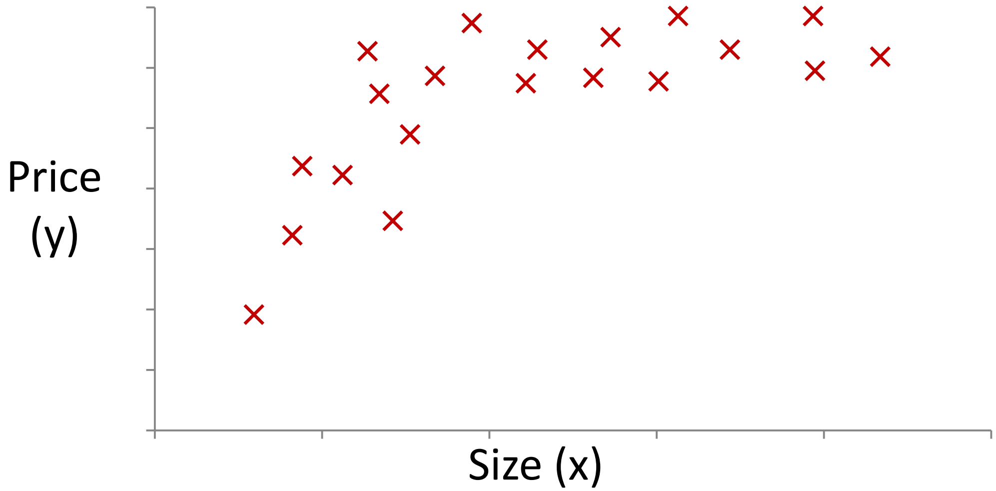
- Too many features (e.g. $m \leq n$).
  - Delete some features, or use regularization.

Features vs variables

Housing prices prediction

$$h_{\beta}(x) = \beta_0 + \beta_1 frontage + \cdots + \beta_p depth$$

Polynomial regression



$$\beta_0 + \beta_1 x + \beta_2 x^2$$

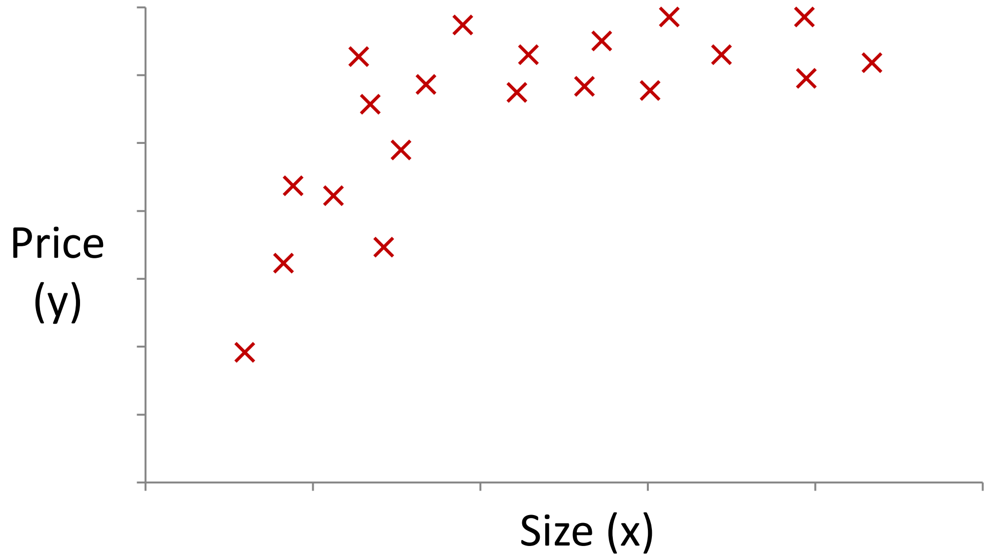
$$\beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

$$h_\beta(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

$$\beta_0 + \beta_1 (size) + \beta_2 (size)^2 + \beta_3 (size)^3$$

$$x_1 = (size)$$
$$x_2 = (size)^2$$
$$x_3 = (size)^3$$

Choice of features



$$h_{\beta}(x) = \beta_0 + \beta_1(size) + \beta_2(size)^2$$

$$h_{\beta}(x) = \beta_0 + \beta_1(size) + \beta_2\sqrt{(size)}$$