Linear Regression with Multiple Variables

KAUSHIK ROY

Center for Brain Inspired Computing

Multiple Features for Linear Regression

➢ Hypothesis and notations

Example: housing price prediction (Lafayette, IN)

Hypothesis: To predict the sale price $\boldsymbol{y}$ of homes

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots + \theta_n x_n$$

(n is total number of features)

Features:

x_1 = Size ($feet^2$)

x_2 = Number of bedrooms

x_3 = Age of home (years)

⋮

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

For convenience of notation, define $x_0 = 1$

Prediction:

y = Price

Hypothesis expression is simplified to

$h_\theta(x) = \theta^T x$ (T denotes matrix transpose)

C-BRIC

Gradient Descent for Linear Regression

- Gradient descent algorithm

Hypothesis: $h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Parameters: $\theta_0, \theta_1, \theta_2, \cdots \theta_n$

Cost function: $J(\theta_0, \theta_1,\ \theta_2, \cdots \theta_n) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$ $\quad x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix}$

in which $(1 \leq i \leq m)$

($\boldsymbol{n}$ is the number of features and $\boldsymbol{m}$ is the number of training samples)

Gradient descent:

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1,\ \theta_2, \cdots \theta_n) \quad (\alpha \text{ is learning rate})$$

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

} (simultaneously update for every $j = 0, 1, \ldots n$)

C-BRIC

Gradient Descent for Linear Regression

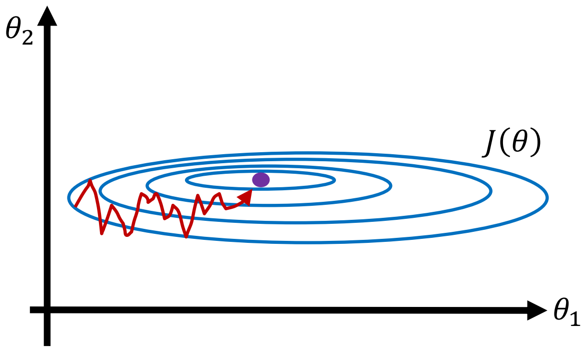
➢ Gradient descent in practice I: Feature Scaling

Idea: Make sure features are on a similar scale, so that gradient descent can converge more quickly.

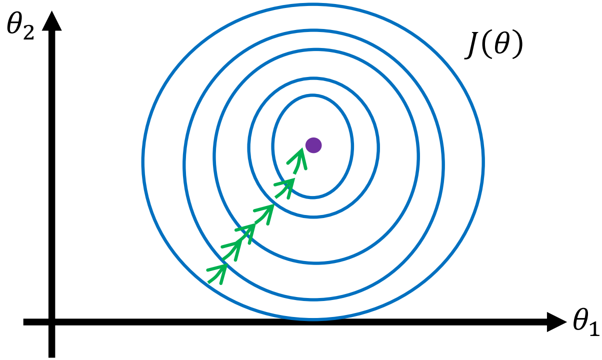
E.g. housing prices prediction

$$x_1 = size\ (0 - 2000\ feet^2)$$
$$x_2 = number\ of\ bedrooms\ (1 - 5)$$

$$x_1 = \frac{size\ (feet^2)}{2000} \quad (0 \le x_1 \le 1)$$
$$x_2 = \frac{number\ of\ bedrooms}{5} \quad (0 \le x_2 \le 1)$$



Very skewed contour of cost function $J(\theta)$ in feature space makes it very difficult for gradient descent to quickly find the global minima.



Gradient descent can find a more direct path to the global minima after feature scaling.

Gradient Descent for Linear Regression

➢ Gradient descent in practice I: Feature Scaling

Feature Scaling

- Mean normalization

Replace x_i with $x_i - \mu_i$ to make features have approximately zero mean

(Do not apply to $x_0 = 1$)

E.g. $x_i \leftarrow \frac{x_i - \mu_i}{S_i}$

μ_i: Average value of x_i in the training set

S_i: range of value (max-min) or standard deviation of x_i in the training set.

C-BRIC

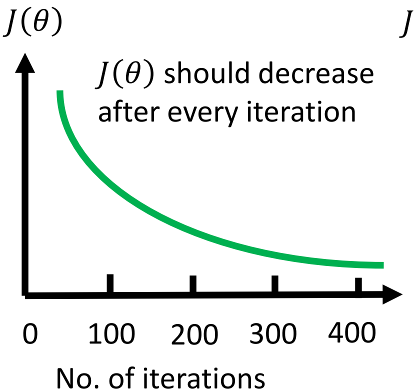
Gradient Descent for Linear Regression

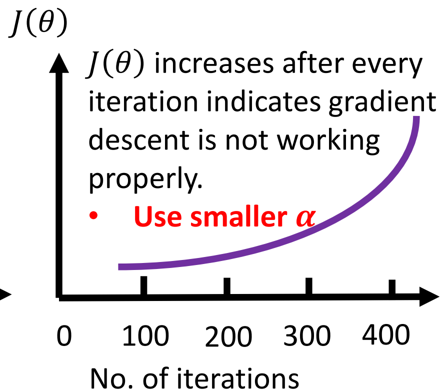
➢ Gradient descent in practice II: Learning rate

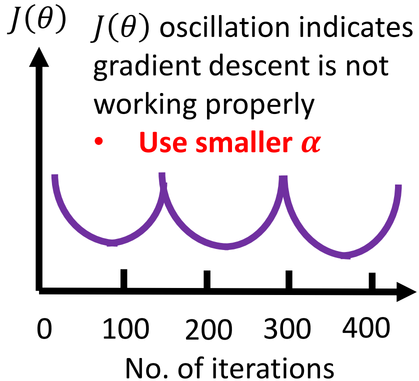
Gradient descent

$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$ (α is learning rate)

- “Debugging”: How to make sure gradient descent is working correctly ?
- How to choose learning rate α ?







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

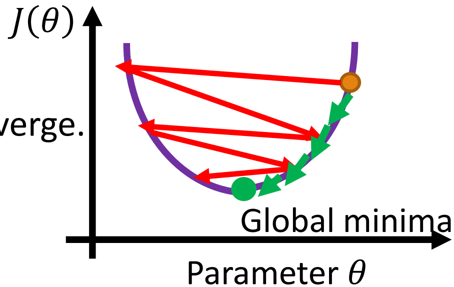
C-BRIC

Gradient Descent for Linear Regression

➢ Gradient descent in practice II: Learning rate

Gradient descent

$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$ (α is learning rate)



If α is too small: slow convergence.
If α is too large: $J(\theta)$ may oscillate and may not converge.
To choose α, try

$\cdots, 0.001, \quad 0.01, \quad 0.1, \quad 1, \cdots$

and plot $J(\theta)$ vs # iterations figure to determine the proper α to use.

Choose the largest possible α (or slightly smaller value) as the learning rate for gradient descent.

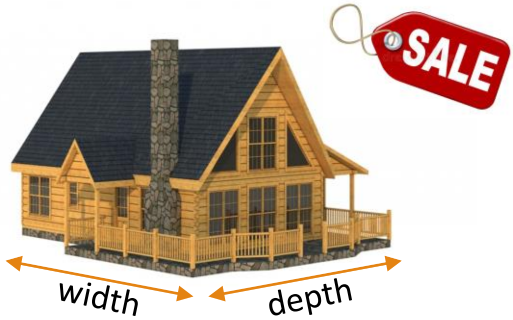
Features and Polynomial Regression

- Feature selection

Selecting features
$width\ W, depth\ D$

Learning models
Model 1:
$h_\theta(x) = \theta_0 + \theta_1 \times W + \theta_2 \times D$

Model 2:
Define new feature area: $A = W \times D$
$h_\theta(x) = \theta_0 + \theta_1 \times A$



Features:
$W = width\ (feet)$
$D = depth\ (feet)$

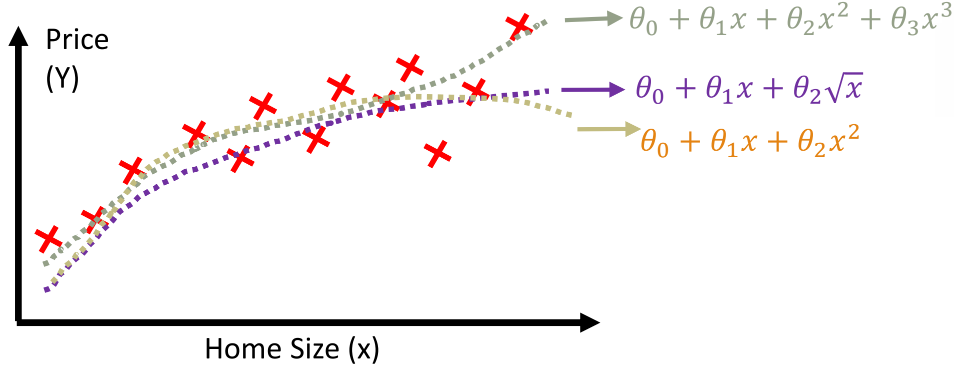
Prediction:
$y = \text{Price}$

Choice of different features lead to different learning models, and powerful models can be built by choosing appropriate features

C-BRIC

Features and Polynomial Regression

➢ Polynomial regression



Features:
$x = size\ (feet^2)$

Prediction:
$y =$ Price

$$
\begin{aligned}
h_\theta(x) &= \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 \quad (1 \le x \le 1{,}000) \\
&= \theta_0 + \theta_1 (size) + \theta_2 (size)^2 + \theta_3 (size)^3
\end{aligned}
$$

$$
\begin{aligned}
x_1 &= (size) && (1 \le x_1 \le 1{,}000) \\
x_2 &= (size)^2 && (1 \le x_2 \le 1{,}000{,}000) \\
x_3 &= (size)^3 && (1 \le x_3 \le 1{,}000{,}000{,}000)
\end{aligned}
$$

Note: Feature scaling is required for gradient descent to work efficiently due to drastic variance in feature ranges.

C-BRIC