Lecture 4: Model parameter estimation - gradient

Wengang Mao (Marine Technology)
Department of Mechanics and Maritime Sciences,
Chalmers University of Technology,
Goteborg, Sweden

1

Contents of this lecture

- Basic meaning of the model regression
- Define the regression (model optimization) problem
- Estimation of the regression model (parameters)
  - Euler-Lagrange theorem (mathematically explicit solution)
  - Gradient descent algorithm (numerical approximation)
- Numerical method: the gradient descent algorithms
- Computer examples

2

Model regression (1)

- **For parametric regression models, let write them in a more general form**
  - $\hat{Y} = E[Y|\boldsymbol{X}] = \beta_0 + \beta_1 f(X_1) + \beta_2 f(X_2) + \cdots$
  - $f(X_i), i = 1,2, \ldots$, represent deterministic transformation of X, such as X_n, log(X), exp(X),…
  - The regression is to find optimal values of β_0, β_1,…, to minimize cost function of optimization

Model regression (2)

- **For the nonparametric regression models (smoothing moving average)**

$$\hat{Y} = E[Y|X] = \sum_{i=1}^{k} g(x - x_i) y_i$$

  - Which kernels $g(x)$ to choose for the smoothing, e.g., $g(x)$ can be normal, box, or other functions?
  - What parameters to choose for a picked kernel $g(x)$?
  - What is the width of the smooth, i.e., k?

Model regression problem def.

The procedure for the general model regression is

Data (n obs.): $\left(y^{(1)}, x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \dots\right), \left(y^{(2)}, x_1^{(2)}, x_2^{(2)}, x_3^{(2)}, \dots\right), \dots, \left(y^{(n)}, x_1^{(n)}, x_2^{(n)}, x_3^{(n)}, \dots\right)$

Hypothesis: $\hat{f}(\boldsymbol{X}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots$

Parameters: $\beta_0, \beta_1, \beta_2, \dots$

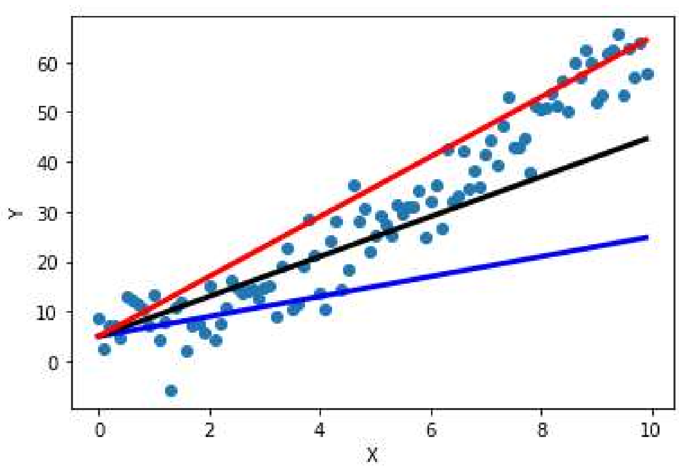
Cost Function: $J(\beta_0, \beta_1, \dots) = \frac{1}{2n}\sum_{i=1}^{n}(\hat{f}(\boldsymbol{X}^{(i)}) - y^{(i)})^2$

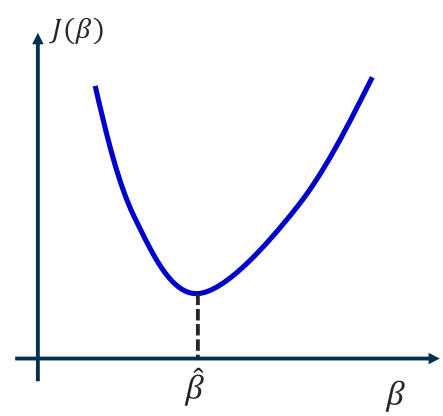
OLS, Ridge, Lasso

Goal: $\underset{\beta_0, \beta_1, \dots}{\text{minimize}}\, J(\beta_0, \beta_1, \dots)$

Regression: parameter estimation

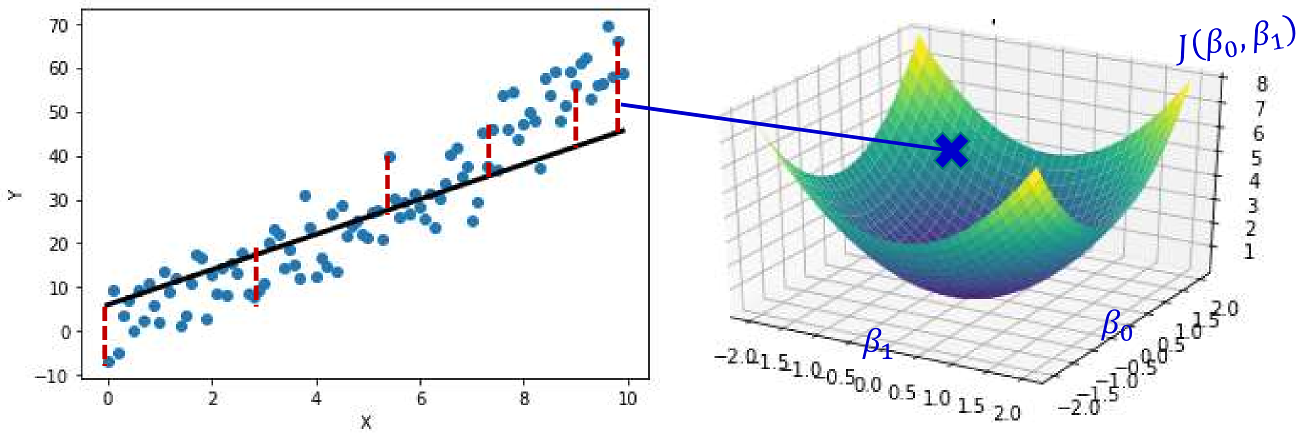
- For the cost function $J(\beta)$, the regression is to find $\hat{\beta}$ that can minimize J





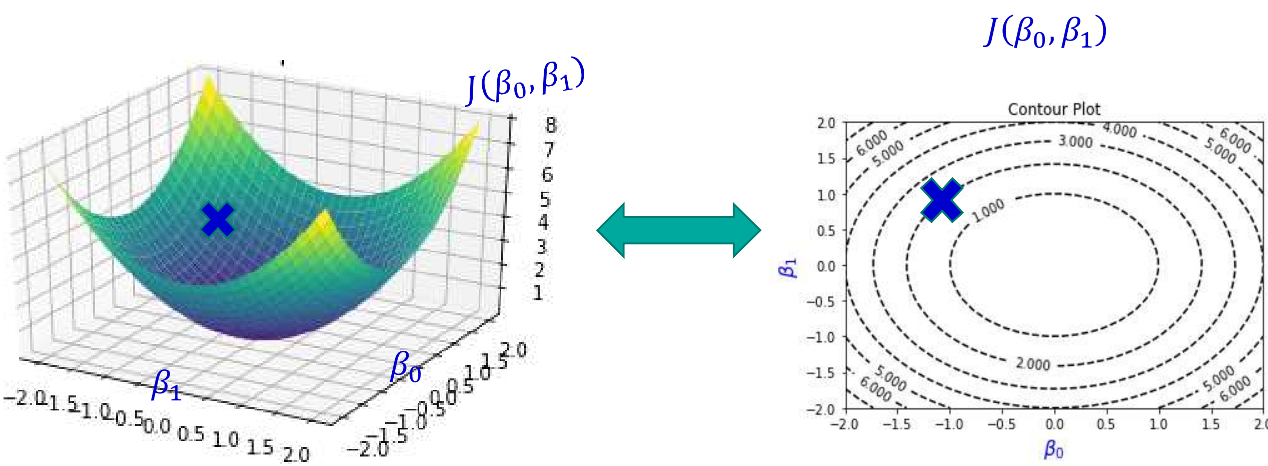
Regression: parameter estimation

- For the cost function $J(\beta_0, \beta_1)$, the regression is to find $\hat{\beta}_0$ and $\hat{\beta}_1$ that can minimize J



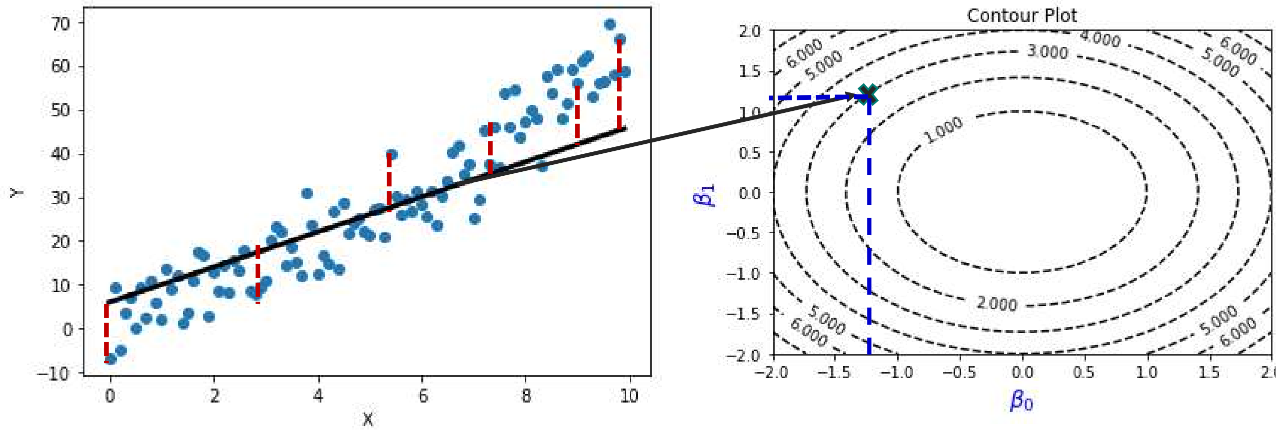
Regression: parameter estimation

- For the cost function $J(\beta_0, \beta_1)$, the regression is to find $\hat{\beta}_0$ and $\hat{\beta}_1$ that can minimize J



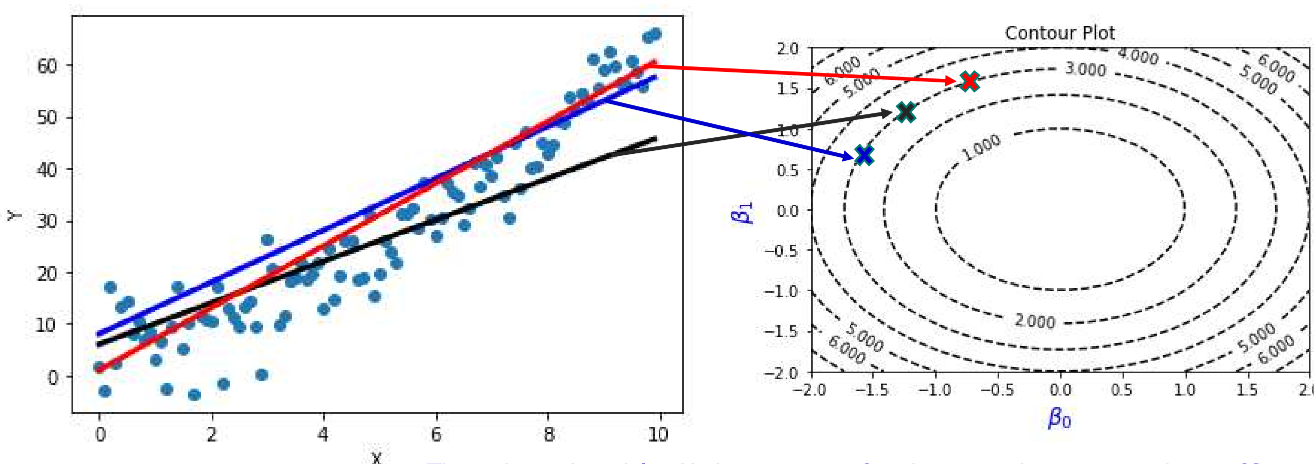
Regression: parameter estimation

For a model $Y = \beta_0 + \beta_1 X$, with data (x_i, y_i) and cost function J as follows:



Regression: parameter estimation

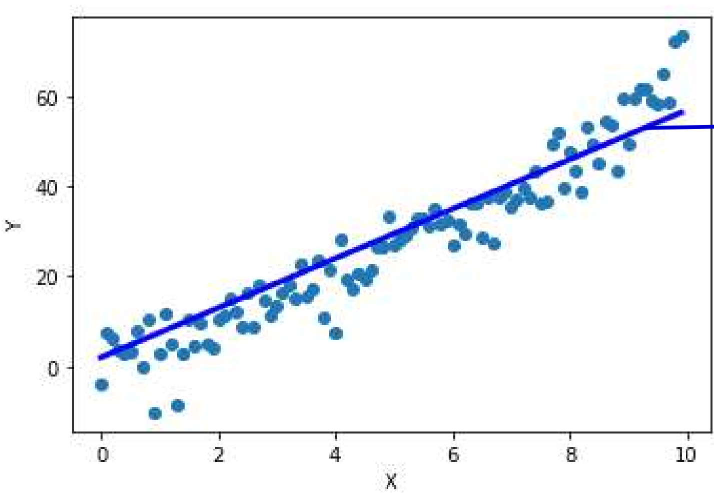
For a model $Y = \beta_0 + \beta_1 X$, with data (x_i, y_i) and cost function J as follows:

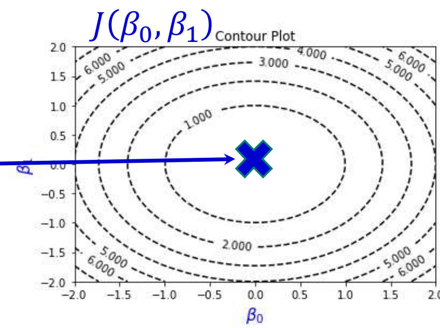


The estimated models with the same cost function mean the same goodness of fit.
Their values are shown as marks on the same contour line.

Regression: parameter estimation

- For the cost function $J(\beta_0, \beta_1)$, the regression is to find $\hat{\beta}_0$ and $\hat{\beta}_1$ that can minimize J





According to the Lagrange multiplier theorem, the solution of parameters to minimize the cost function J is set:

$$\frac{\partial J(\beta_0, \beta_1)}{\partial \beta_0} = 0 \qquad \frac{\partial J(\beta_0, \beta_1)}{\partial \beta_1} = 0$$

Regression: parameter estimation (1)

- When the cost function is very complex, it means that the mathematical solution to the differential equations with respect to parameters is not that straightforward.
- Some numerical methods, e.g., implementation of the gradient descent, could be used to get the parameters of a model to minimize the cost function for the regression
- Let the cost function represented by $J(\beta_0, \beta_1)$, the procedure to get β_0, β_1 minimizing J is:

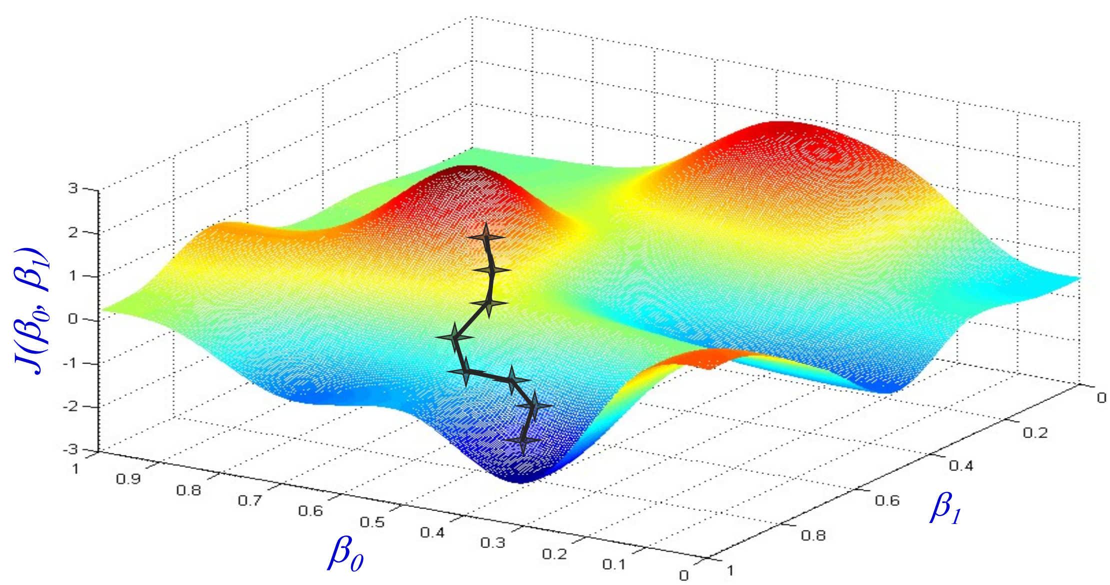
Regression: parameter estimation (2)

- **Workflow for gradient descent for numerical approximation:**
  - Start with some initial values ($\beta_0 = \beta_{0,0}, \beta_1 = \beta_{1,0}$)
  - Updating the values of (β_0, β_1) iteratively according to the gradient of cost functions until the cost function reaches to a minimum point (not always successful for global minimum)

13 2023-04-11

13

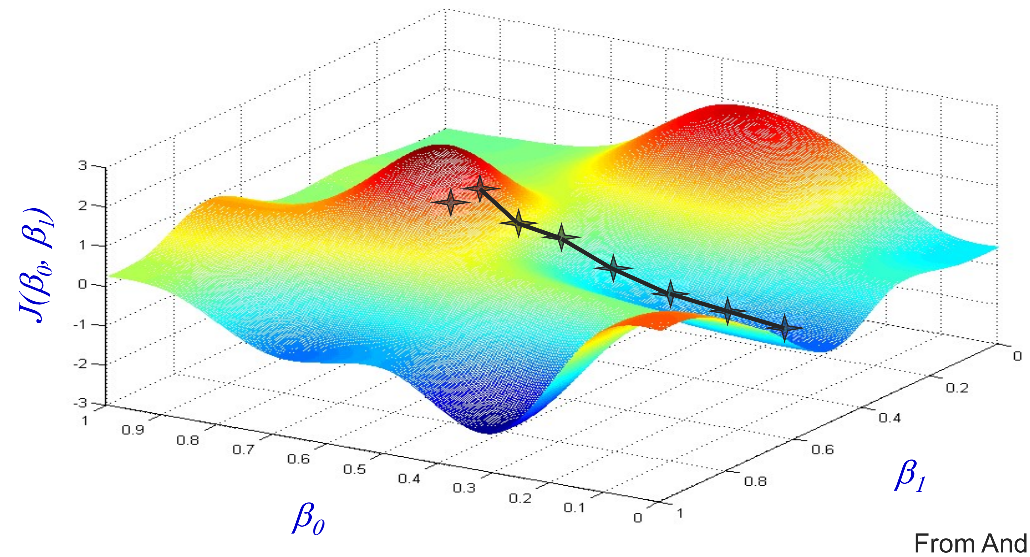
Gradient descent for approximation



From Andrew Ng's lecture notes

14

Initial values matter for approximation



From Andrew Ng's lecture notes

15

(Batch) Gradient descent algorithm

Procedures to implement the gradient descent algorithm

1. Select initial values of (β_0, β_1) according to your experiences, i.e., $(\beta_{0,0}, \beta_{1,0})$
2. Choose a learning rate coefficient α
3. Estimate the gradient of the cost function J at the values of $(\beta_{0,0}, \beta_{1,0})$, i.e.,
   $\frac{\partial}{\partial \beta_0} J(\beta_{0,0}, \beta_{1,0})$, and $\frac{\partial}{\partial \beta_1} J(\beta_{0,0}, \beta_{1,0})$
4. Update all the model parameters simultaneously according to the learning rate
   $\beta_{0,1} = \beta_{0,0} - \alpha \frac{\partial}{\partial \beta_0} J(\beta_{0,0}, \beta_{1,0})$,
   $\beta_{1,1} = \beta_{1,0} - \alpha \frac{\partial}{\partial \beta_1} J(\beta_{0,0}, \beta_{1,0})$
5. Repeat steps (3-4) until the cost function J convergence to a minimum point

2023-04-11

16

Stochastic Gradient descent Algor.

Procedures to implement the gradient descent algorithm

1. Select initial values of (β_0, β_1) according to your experiences, i.e., $(\beta_{0,0}, \beta_{1,0})$
2. Choose a learning rate coefficient α
3. Estimate the error of the first instance (data point), i.e., $\varepsilon = y_i - \beta_{0,0} - \beta_{1,0}X_i$
4. Update all the model parameters simultaneously according to the learning rate

   $\beta_{0,1} = \beta_{0,0} - \alpha \times \varepsilon,$

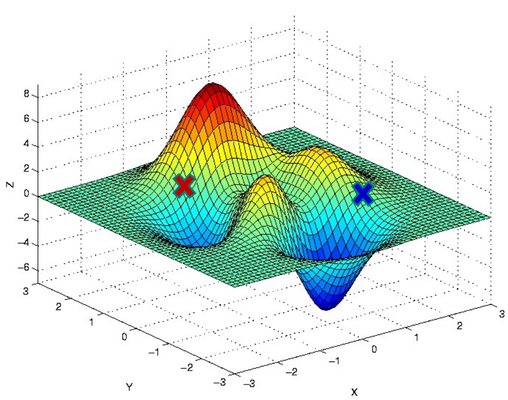
   $\beta_{1,1} = \beta_{1,0} - \alpha \times \varepsilon \times X_i,$
5. Repeat steps (3-4) until the parameters converge to stable parameters

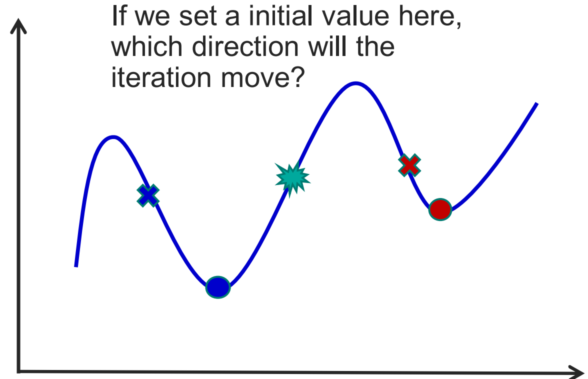
2023-04-11

17

Key elements in Gradient algorithm

- **Initial values of the model parameters are important for the convergence study**
- **Not necessarily always convergence to a global minimum cost value**



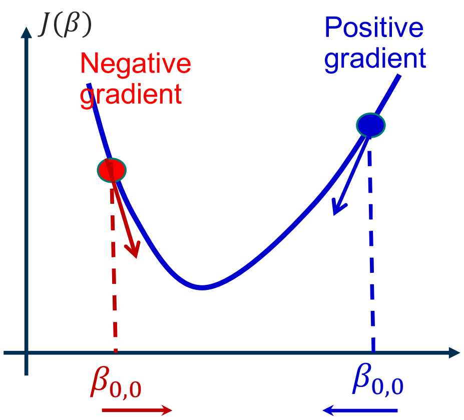


18 2023-04-11

18

Key elements in Gradient algorithm

- Movement of parameter update in continuous iterations



$$\beta_{0,1} = \beta_{0,0} - \alpha \frac{\partial}{\partial \beta_0} J(\beta_{0,0}, \beta_{1,0}),$$

Let the learning rate α as a positive value

- On the negative side, since the derivative is negative, the parameter increments will be positive, i.e., move right
- On the negative side, the parameter increments will move Left
- So, on both side, the parameters will move toward the minimum location of the cost function

19 2023-04-11

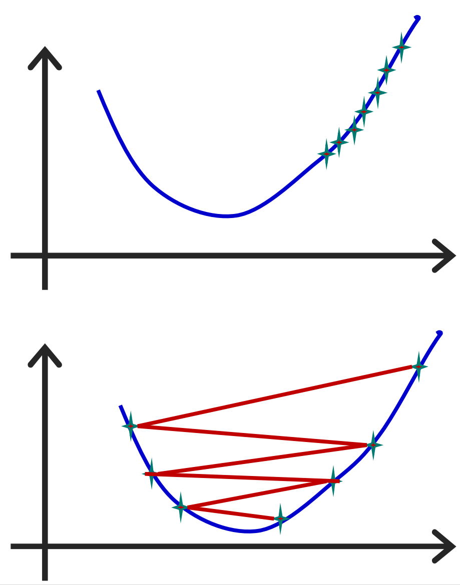
19

Key elements in Gradient algorithm

The results will be sensitive to values of the learning rate:

- If α is too small, the convergence of gradient descent will be slow
- If α is too large, the convergence can overshoot the minimum, or even diverge

$$\beta_{0,1} = \beta_{0,0} - \alpha \frac{\partial}{\partial \beta_0} J(\beta_{0,0}, \beta_{1,0}),$$

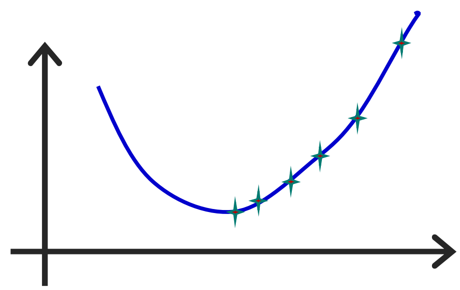


20 2023-04-11

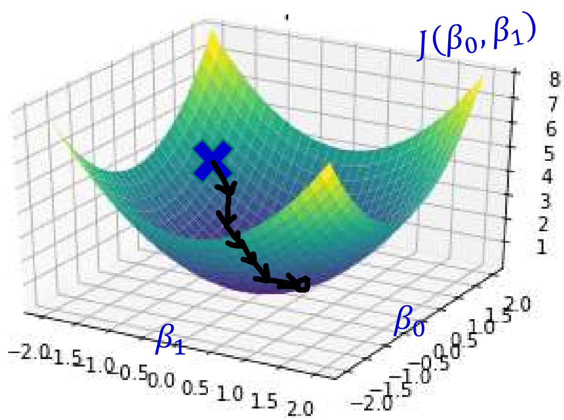
20

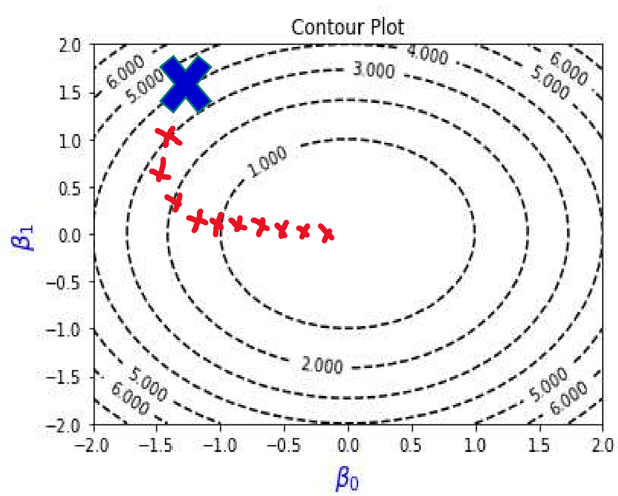
Key elements in Gradient algorithm

- Should the value of α be adjusted (reduced) in each iteration when the cost function approaches to its minimum location?
- *Probably not, because gradient descent can automatically help the iteration take smaller steps, because the derivative also decreases.*



Gradient descent for multivariate regression





Final remarks on gradient descent

- For each iteration of the gradient descent method, all the data should be used to estimate the cost for the pre-assumed model parameters.
- The gradient descents ideas have been also widely used in other machine learning algorithms.
- For example, to be combined with the boosting method, the so-called XG boost method is one of the most powerful ML algorithm for the model estimation.

Introduction of assignment project 2

- Prediction of ship power consumption in terms of other parameters

CHALMERS
UNIVERSITY OF TECHNOLOGY

25