情報工学実験4：データマイニング班

(week 3) 線形回帰モデルと最急降下法

1. モデルとは？（問題設定、アルゴリズム、モデル）
2. 線形回帰モデル
3. 仮説、損失関数、目的関数
4. 最小二乗法
5. 最急降下法
6. 参考サイト

実験ページ: http://ie.u-ryukyu.ac.jp/~tnal/2014/info4/dm/

Problems, Models, Algorithms

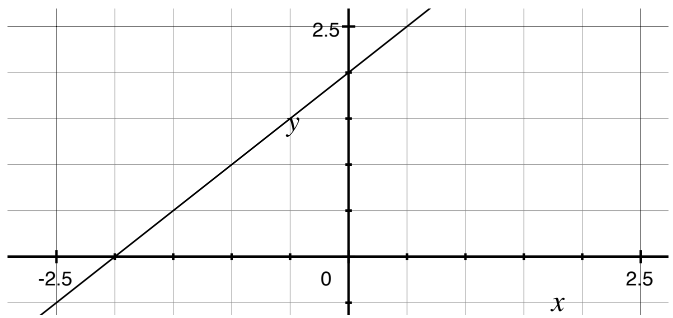
- Problems
  - Classification
  - Regression
  - Clustering
- Algorithms
  - Ordinary Least Squares
  - Gradient Descent
  - Back Propagation

What is that?

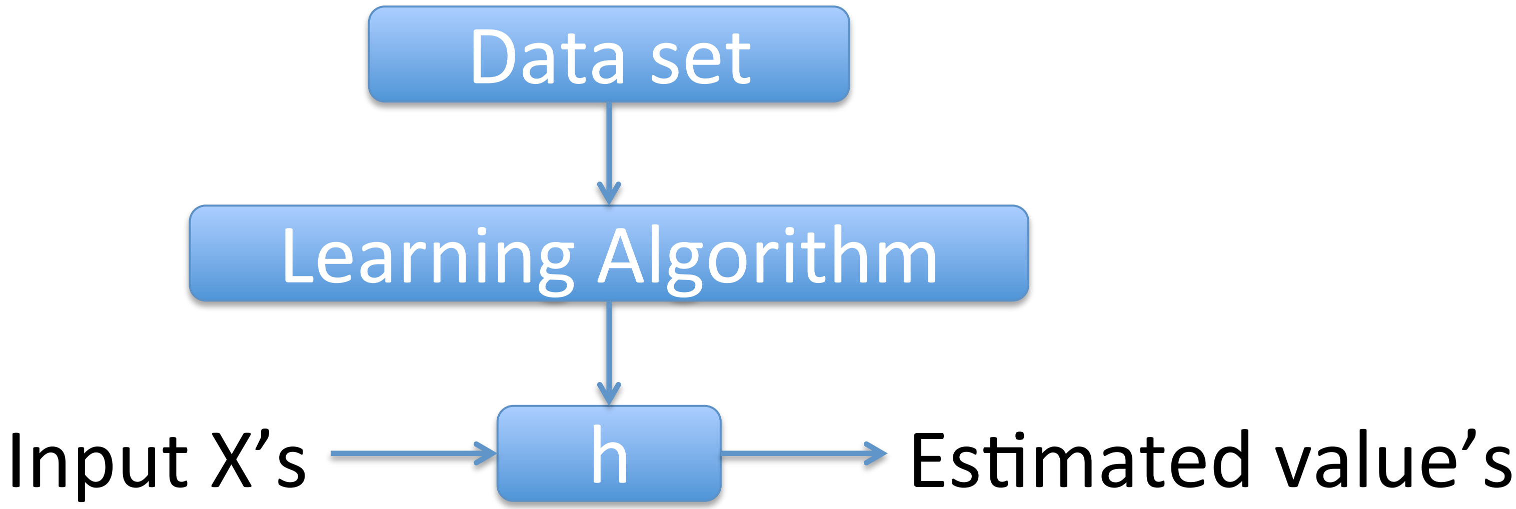
- Models
  - Linear Regression Model
  - Generalized Linear Models
  - Neural Network
  - Decision Tree
  - (other kinds of models)
    - Bag-of-words document model

Models

- Represent by any formulas with (sometimes one) parameters for the relationship between input X's and output Y's.
  - In machine learning, the formulas called as "hypothesis".
  - E.g., h = a*x + b
    - a, b: parameters
  - Parameterized model.
  - Predictive model. (a=1, b=2)



Problem <-> Algorithm + Model



Linear Regression Model

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 = \sum \theta_i x_i = \sum \theta_i \Phi_i(x)$$

$$h_\theta(x) = \theta_0 + \theta_1 x$$

- How do we prepare a model?
- How do we evaluate the goodness?
- How do we choose the appropriate parameters?

Linear Regression Model

- Training datasets
  - (x,y) = (4,7), (8,10), (13,11), (17,14)
- Hypothesis

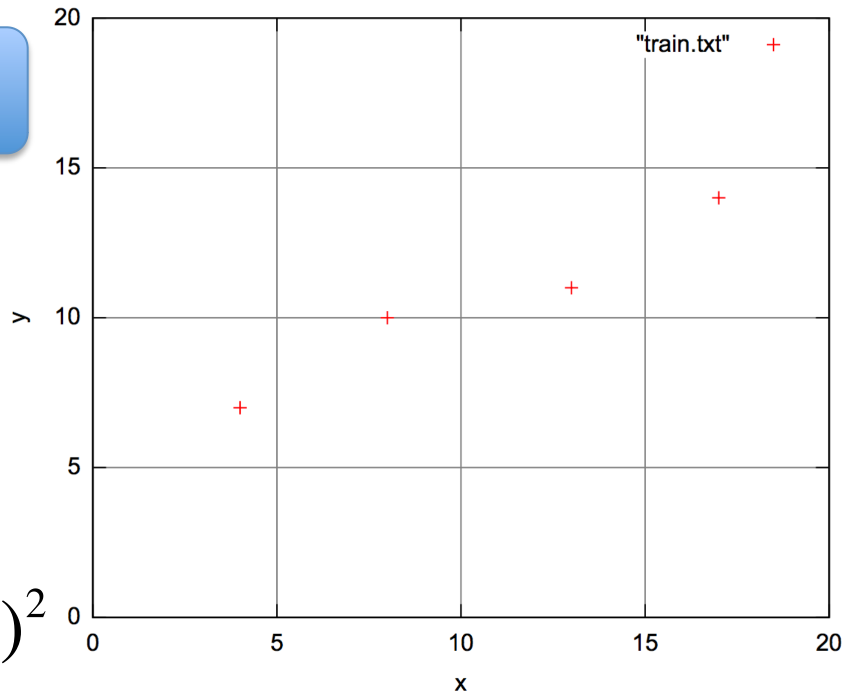
$$h_\theta(x) = \theta_0 + \theta_1 x$$

Assumption 1
Linear function

- Parameters
  - θ0, θ1
- Cost function

Assumption 2
Squared error

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$



- Objective function (measurement of the goodness)

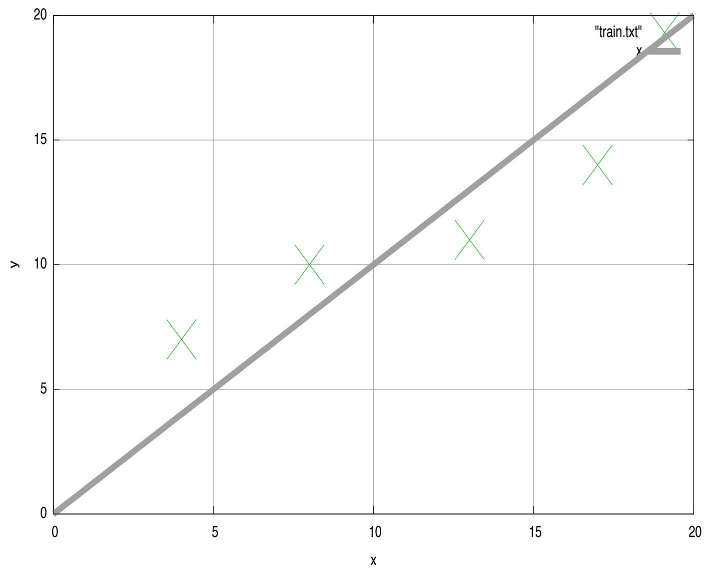
$$\min_{\theta} J(\theta_0, \theta_1)$$

Hypothesis vs. Cost function (θ1=1)

θ0=0, θ1=1, (x,y)=(4,7), (8,10), (13,11)

Hypothesis:

$$h(x) = x$$

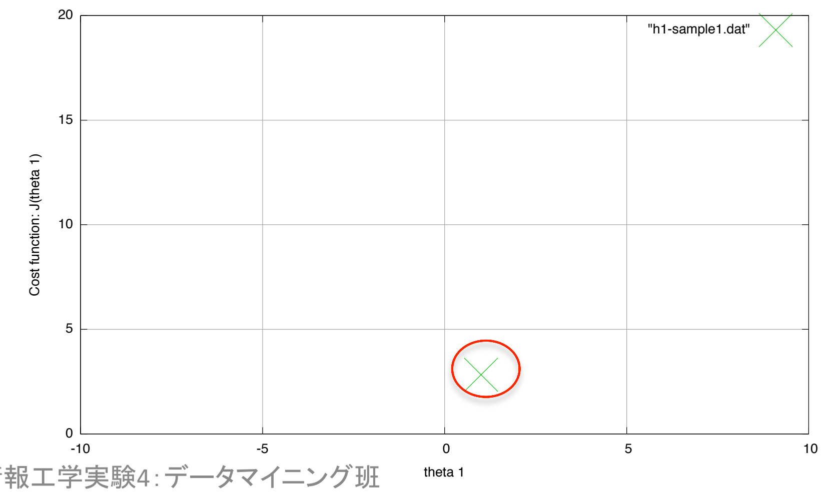


Cost function:

$$J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$J(1) = \frac{1}{2m}((4-7)^2 + (8-10)^2 + (13-11)^2)$$

$$J(1) = \frac{1}{2*3}(9+4+4) = \frac{17}{6} = 2.83$$

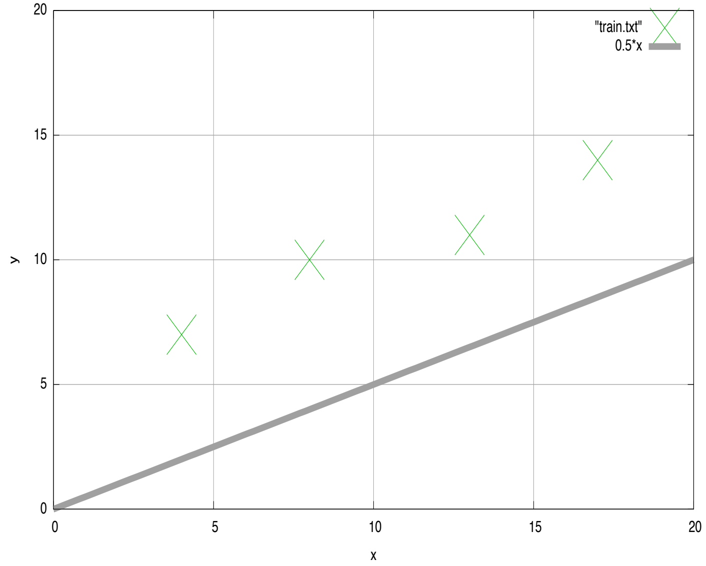


Hypothesis vs. Cost function (θ1=0.5)

θ0=0, θ1=0.5, (x,y)=(4,7), (8,10), (13,11)

Hypothesis:

$$h(x) = 0.5 * x$$

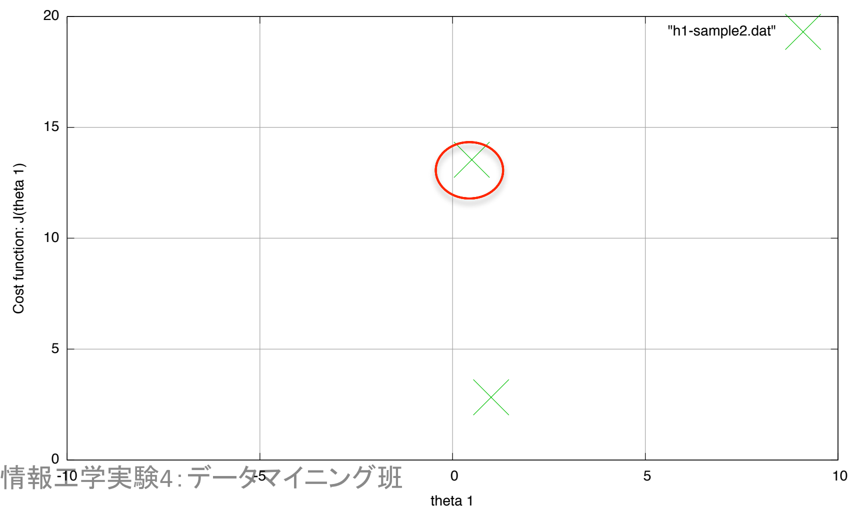


Cost function:

$$J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$J(0.5) = \frac{1}{2m}((2-7)^2 + (4-10)^2 + (6.5-11)^2)$$

$$J(0.5) = \frac{1}{2*3}(25+36+20.25) = \frac{81.25}{6} = 13.54$$



Hypothesis vs. Cost function (θ1=others)

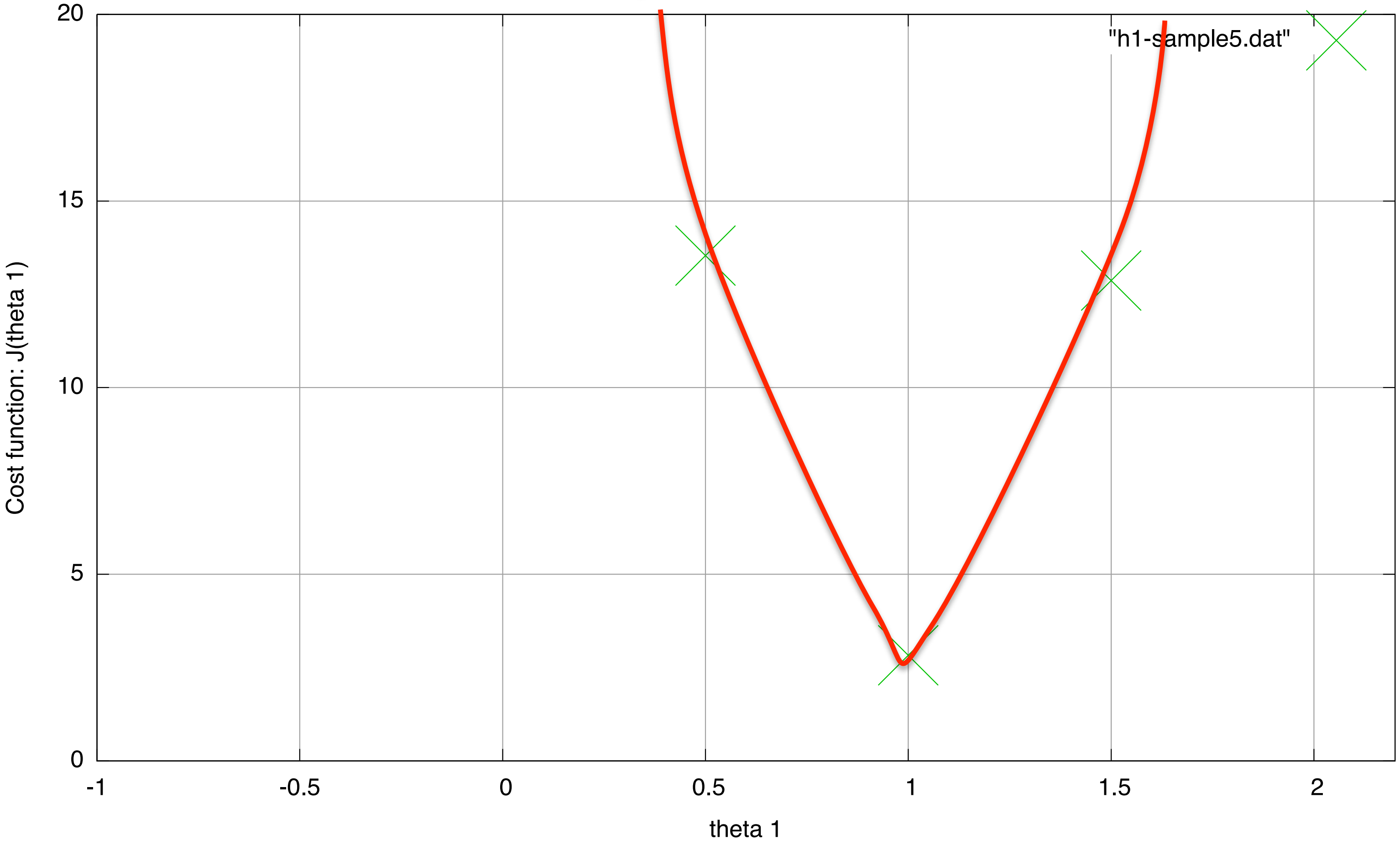
θ0=0, θ1=others, (x,y)=(4,7), (8,10), (13,11)

- θ=0:
  - H(x)=0*x=0
  - J(0)=1/6{(0-4)^2+(0-10)^2+(0-13)^2}
    - =1/6{16+100+169}=47.5
- θ=2:
  - H(x)=2*x
  - J(2)=1/6{(8-7)^2+(16-10)^2+(26-11)^2}
    - =1/6{1+25+225}=41.83
- θ=1.5:
  - H(x)=1.5*x
  - J(1.5)=1/6{(6-7)^2+(12-10)^2+(19.5-11)^2}
    - =1/6{1+4+72.25}=12.87

Objective function: minimize J(θ1)

- How do we observe the shape of function?
- How do we observe the behavior of GD?

Convex function

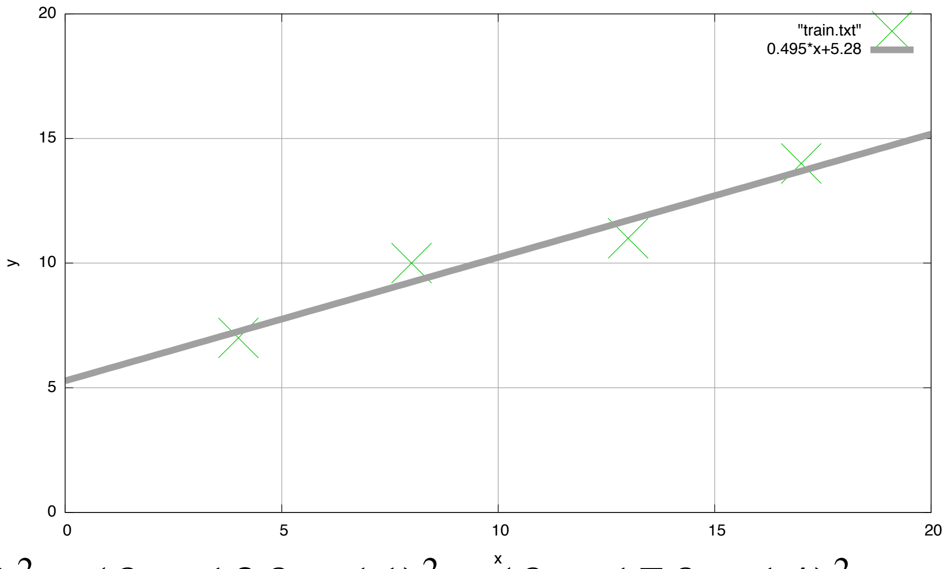


Ordinary Least Squares

problems?

$$h(x) = \theta_0 + \theta_1 x$$ (x,y)=(4,7), (8,10), (13,11), (17,14)

$$7 = \theta_0 + 4\theta_1$$
$$0 = \theta_0 + 4\theta_1 - 7$$
$$e_1 := \theta_0 + 4\theta_1 - 7$$
$$e_1^2 = (\theta_0 + 4\theta_1 - 7)^2$$



$$E = \sum e_i^2 \geq 0$$
$$= (\theta_0 + 4\theta_1 - 7)^2 + (\theta_0 + 8\theta_1 - 10)^2 + (\theta_0 + 13\theta_1 - 11)^2 + (\theta_0 + 17\theta_1 - 14)^2$$
$$= 538\theta_1^2 + 84\theta_0\theta_1 + 4\theta_0^2 - 978\theta_1 - 84\theta_0 + 466$$
$$= (2\theta_1 + 21\theta_0 - 21)^2 + 97(\theta_0 - 48/97)^2 + 121/97$$

θ0=1029/194≒5.28, θ1=48/97≒0.495
h(x) = 5.28 + 0.495x

Ref., http://gihyo.jp/dev/serial/01/machine-learning/0008

Gradient descent algorithm

Repeat until convergence {

$$\theta_i := \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta_0, \theta_1)$$

}

(1) Start with any parameters.
(2) Update the parameters simultaneously, until convergence.

<u>Simple example</u>

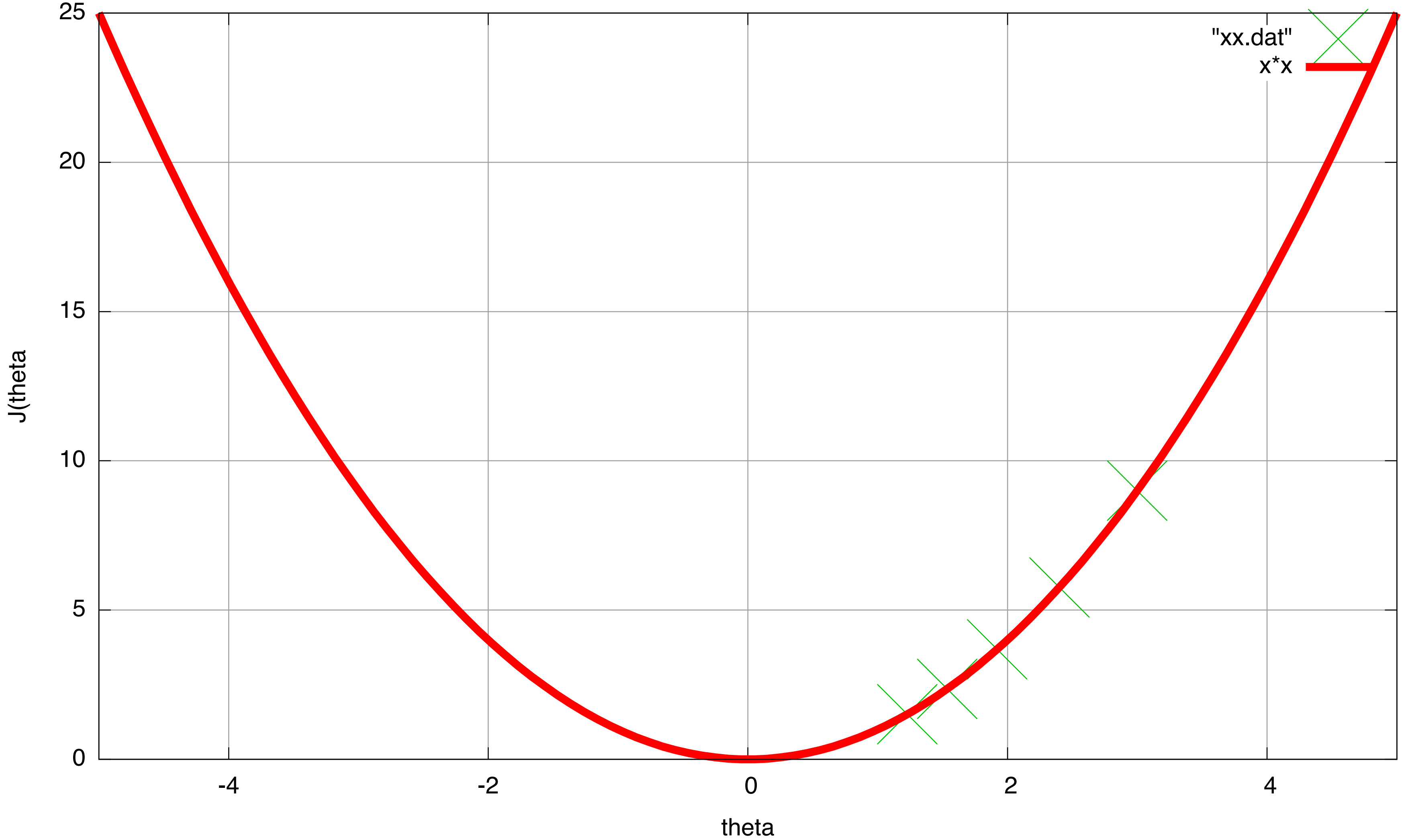
$$J(\theta) = \theta^2, \alpha = 0.1$$

$$new\_\theta = \theta - \alpha \frac{d}{d\theta} J(\theta)$$

$$= \theta - 0.1 * 2\theta = \theta - 0.2\theta = 0.8\theta$$

- e.g., α=0. 1, θ=3, J(θ)=9
- 1st update
    - New_θ = 0.8*3 = 2.4
    - J(θ) = 5.76
- 2nd update
    - New_θ = 1.9200000000000004
    - J(θ) = 3.6864000000000012
- 3rd update
    - New_θ = 1.5360000000000005
    - J(θ) = 2.3592960000000014
- 4th update
    - New_θ = 1.2288000000000006
    - J(θ) = 1.5099494400000013

Cont.) the behavior of GD



Gradient descent for Linear Regression

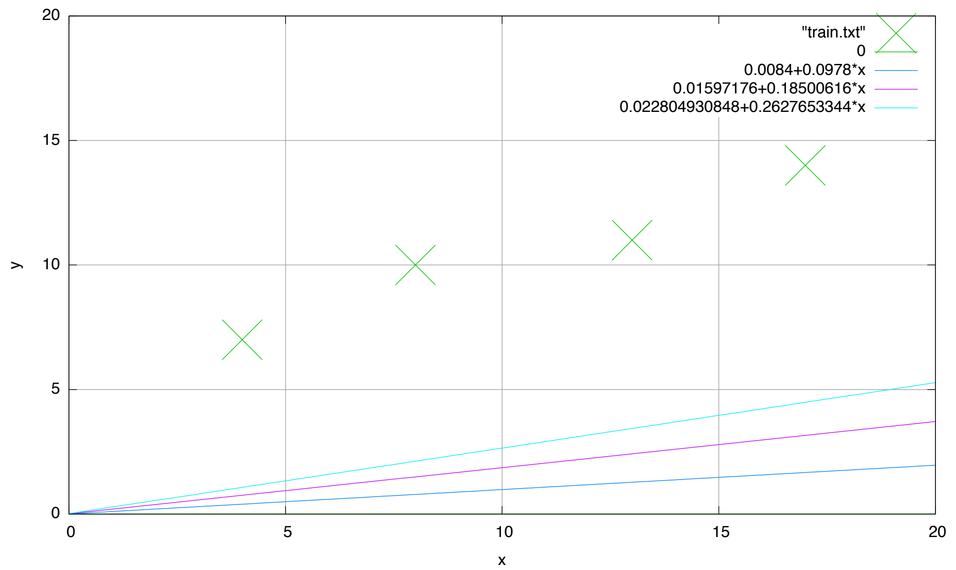
$$\theta_i := \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta_0, \theta_1)$$

$$h_\theta(x) = \theta_0 + \theta_1 x$$ (x,y) = (4,7), (8,10), (13,11), (17,14)

$$J(\theta_0, \theta_1) = 538\theta_1^2 + 84\theta_0\theta_1 + 4\theta_0^2 - 978\theta_1 - 84\theta_0 + 466$$

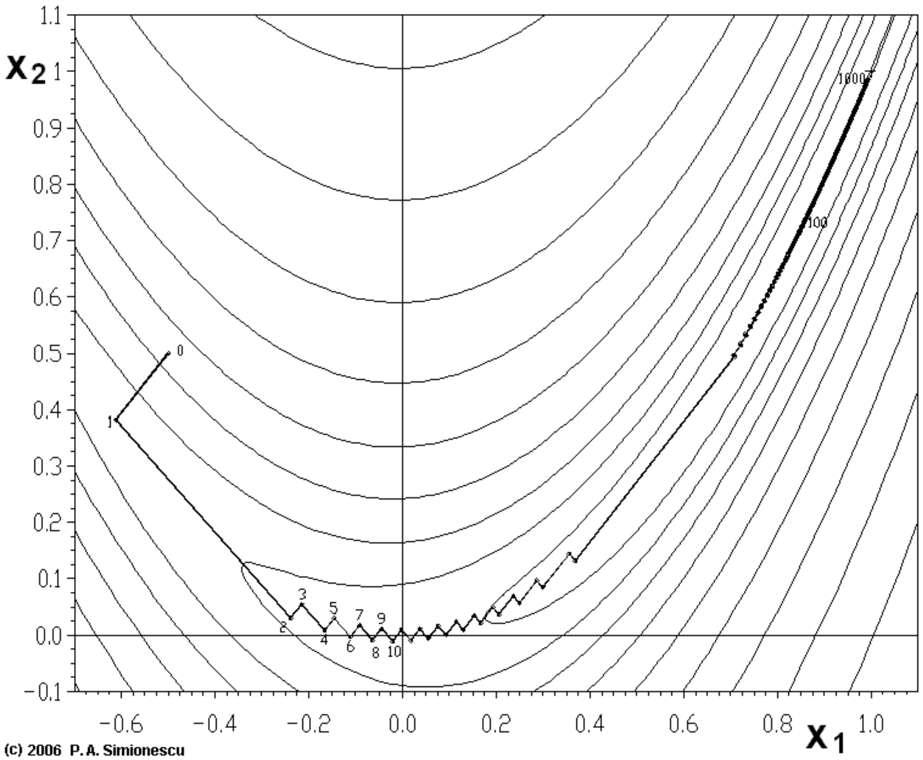
$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = 84\theta_1 + 8\theta_0 - 84$$

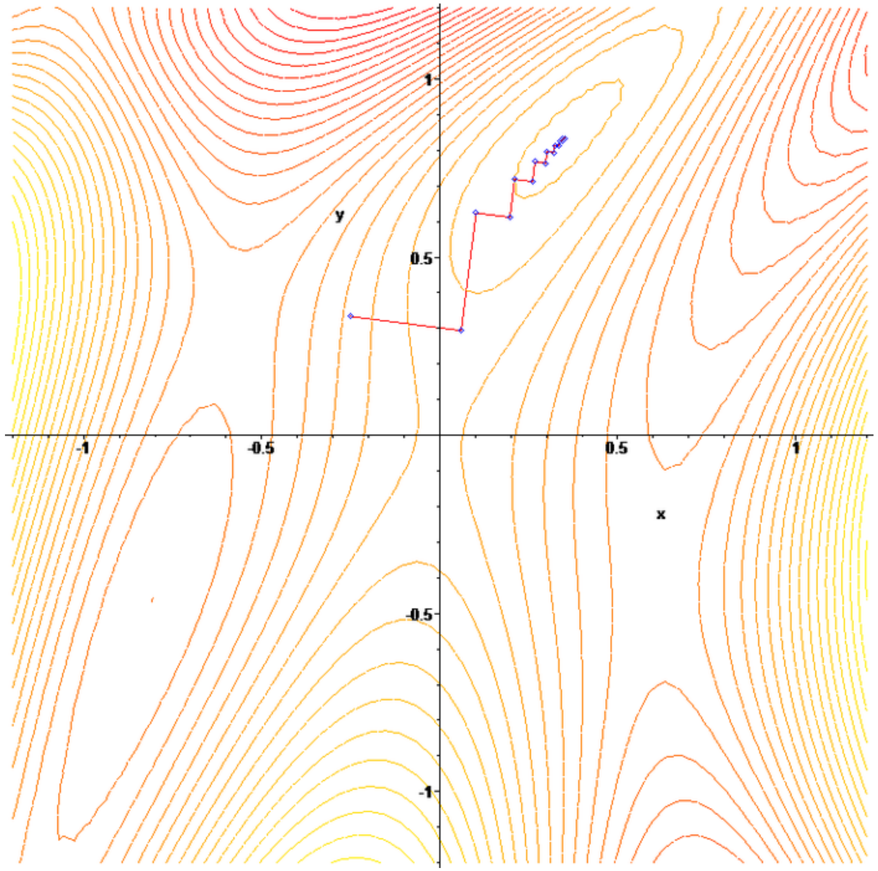
$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = 1076\theta_1 + 84\theta_0 - 978$$



- e.g., α=0.001, θ0=0, θ1=0, J(θ)=466
- 1st update
  - New_θ0 = 0 – 0.001*(-84) = 0.0084
  - New_θ1 = 0 – 0.001*(-978) = 0.0978
  - J(θ) = 374.86117384
- 2nd update
  - New_θ0 = 0.01597176
  - New_θ1 = 0.18500616
  - J(θ) = 302.3858537133122
- 3rd update
  - New_θ0 = 0.022804930848
  - New_θ1 = 0.2627653344
  - J(θ) = 244.7518701063334
- 4th update
  - New_θ0 = 0.028979458094 3616
  - New_θ1 = 0.332100229994368
  - J(θ) = 198.91981002677187

(optional) zig-zagging behavior





Ref., http://en.wikipedia.org/wiki/Gradient_descent

References

- Machine Learning | Coursera, https://class.coursera.org/ml-007
- Gradient descent – Wikipedia, http://en.wikipedia.org/wiki/Gradient_descent
- 数理計画法 第12回, http://www.dais.is.tohoku.ac.jp/~shioura/teaching/mp11/mp11-12.pdf
- 機械学習 はじめよう 第8回　線形回帰[前編], http://gihyo.jp/dev/serial/01/machine-learning/0008
- 機械学習 はじめよう 第9回　線形回帰[後編], http://gihyo.jp/dev/serial/01/machine-learning/0009
- PRMLの線形回帰モデル(線形基底関数モデル), http://www.slideshare.net/yasunoriozaki12/prml-29439402