

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

実験ページ: <http://ie.u-ryukyu.ac.jp/~tnal/2019/info4/dm/>

Example: *Iris* flower data set

review

http://en.wikipedia.org/wiki/Iris_flower_data_set

(1) What is experience E ?

(2) What is task T ?

(3) How to measure the performance P ?

• Classification

– In Classification, the samples belong to two or more classes and we want to learn from already labeled data how to predict the class of unlabeled data.

– E.g., distinguishes the species from each other.

– Dataset = **samples** vs. **features** and **classes**

- Teach data

- supervisory signal

- output data, Y

- target

- 1 class in 3 classes

- Input data, X

- 4 features or attributes

Fisher's *Iris* Data

Sepal length \diamond	Sepal width \diamond	Petal length \diamond	Petal width \diamond	Species \diamond
5.1	3.5	1.4	0.2	<i>I. setosa</i>
4.9	3.0	1.4	0.2	<i>I. setosa</i>
4.7	3.2	1.3	0.2	<i>I. setosa</i>
4.6	3.1	1.5	0.2	<i>I. setosa</i>
5.0	3.6	1.4	0.2	<i>I. setosa</i>

1 sample

Example: boston house prices dataset

<http://archive.ics.uci.edu/ml/datasets/Housing>

review

(1) What is experience E?

(2) What is task T?

(3) How to measure the performance P?

- Regression

- If the desired output consists of one or more continuous variables, then the task is called *regression*.
- E.g., concerns housing values in suburbs of Boston.
- Dataset = **samples** vs. **features** and **continuous variables**

13 features

Continuous variable

CRIM	ZN	INDUS	(中略)	LSTAT	MEDV
6.32E-03	1.80E+01	2.31E+00		4.98E+00	24.00
2.73E-02	0.00E+00	7.07E+00		9.14E+00	21.60
2.73E-02	0.00E+00	7.07E+00		4.03E+00	34.70

1 sample

Example: Overview of clustering methods

<https://scikit-learn.org/stable/modules/clustering.html>

review

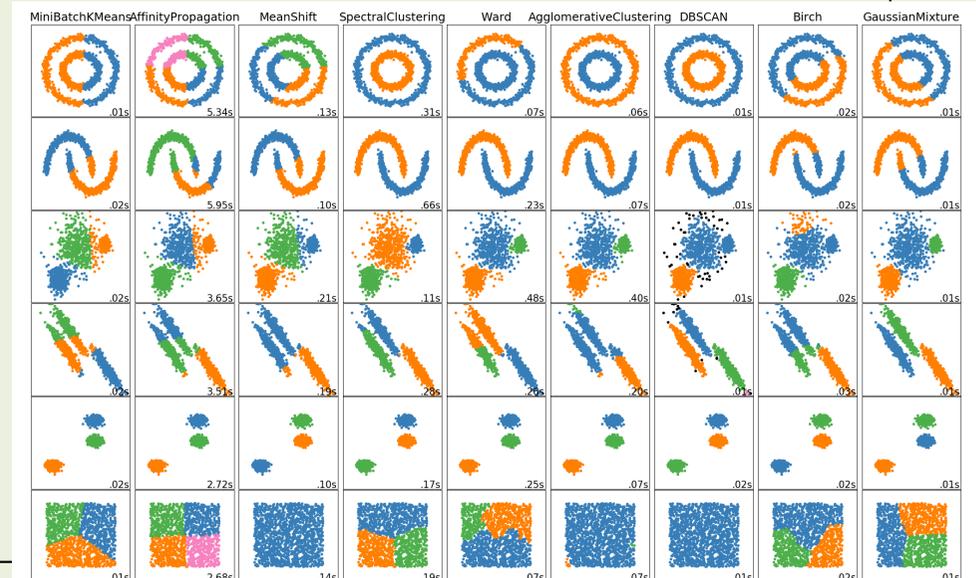
(1) What is experience E?

(2) What is task T?

(3) How to measure the performance P?

• Clustering

- Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense or another) to each other than to those in other groups (clusters).
- Training data consists of a set of input vectors x **without any corresponding target values**.
- Dataset = **samples** vs. **features**



Terminology

review

- ML types
 - supervised, unsupervised, semi-supervised
 - (reinforcement learning, genetic algorithm,,,))
 - Task types
 - classification, regression, clustering
 - sample
 - features, attributes
 - numerical value
 - categorical value
 - true or false
 - supervisory signal, teacher, class, label, target variable
- input, output
 - Input types
 - training data / training set
 - test (for evaluation)
 - validation (for hyper params)
 - model
 - parameters
 - hyper parameters
 - weights, parameters
 - learn, fit
 - predict, estimate
 - evaluation
 - open or close test
 - cross validation

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

An introduction to machine learning with scikit-learn (1/3)

https://github.com/naltoma/intro_jupyter_sklearn

=> intro_sklearn.ipynb

- Loading and an example dataset

- python --version

- Python 3.6.x or 3.7.x

- Example codes

```
from sklearn import datasets
```

```
iris = datasets.load_iris() # datasets.load[tab]
```

```
print(iris.DESCR)
```

```
print(iris.data)
```

```
print(iris.target)
```

```
print(iris.target_names)
```

<http://scikit-learn.org/stable/tutorial/basic/tutorial.html>

An introduction to machine learning with scikit-learn (2/3)

- Learning and predicting

```
from sklearn import svm
```

```
clf = svm.SVC(gamma=0.001, C=100.)
```

```
clf.fit(iris.data[:-1], iris.target[:-1])
```

```
clf.predict(iris.data[-1:])
```

```
# sklearn 0.17以降?, サンプル1個だと書き方に注意。
```

```
print(iris.target[-1])
```

```
clf.score(iris.data, iris.target)
```

<http://scikit-learn.org/stable/tutorial/basic/tutorial.html>

An introduction to machine learning with scikit-learn (3/3)

- Model persistence

- # save

- import pickle

- with open('PredictiveModel.pickle', 'wb') as f:

- pickle.dump(clf, f, pickle.HIGHEST_PROTOCOL)

- # load

- with open('PredictiveModel.pickle', 'rb') as f:

- clf2 = pickle.load(f)

- # check the model

- clf1.predict(iris.data) == clf2.predict(iris.data)

<http://scikit-learn.org/stable/tutorial/basic/tutorial.html>

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

Problems, Models, Algorithms

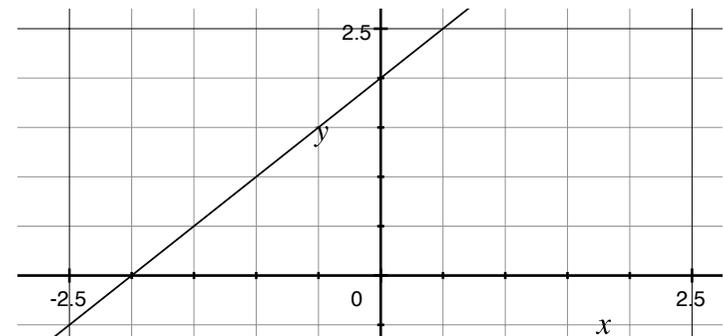
What is that?

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

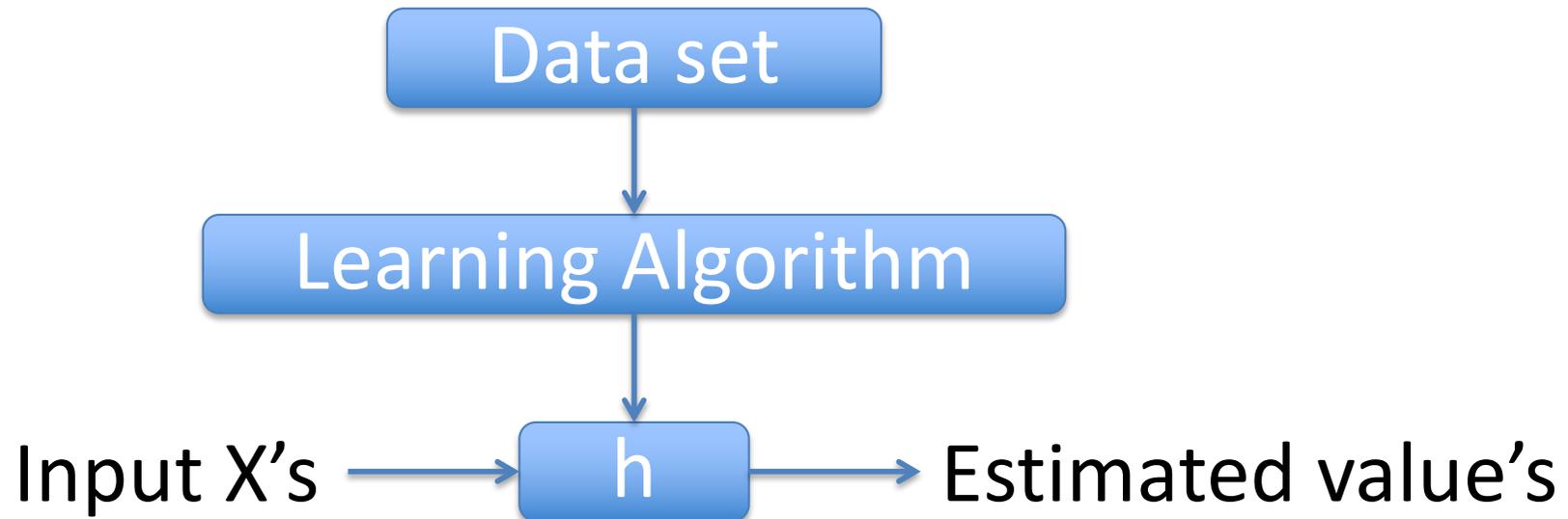
- Models
 - Linear Regression Model
 - Generalized Linear Models
 - Neural Network
 - Decision Tree
 - (other kinds of models)
 - Bag-of-words document model

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. (e.g., $a=1, b=2$)



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?

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

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

$$- \theta_0, \theta_1$$

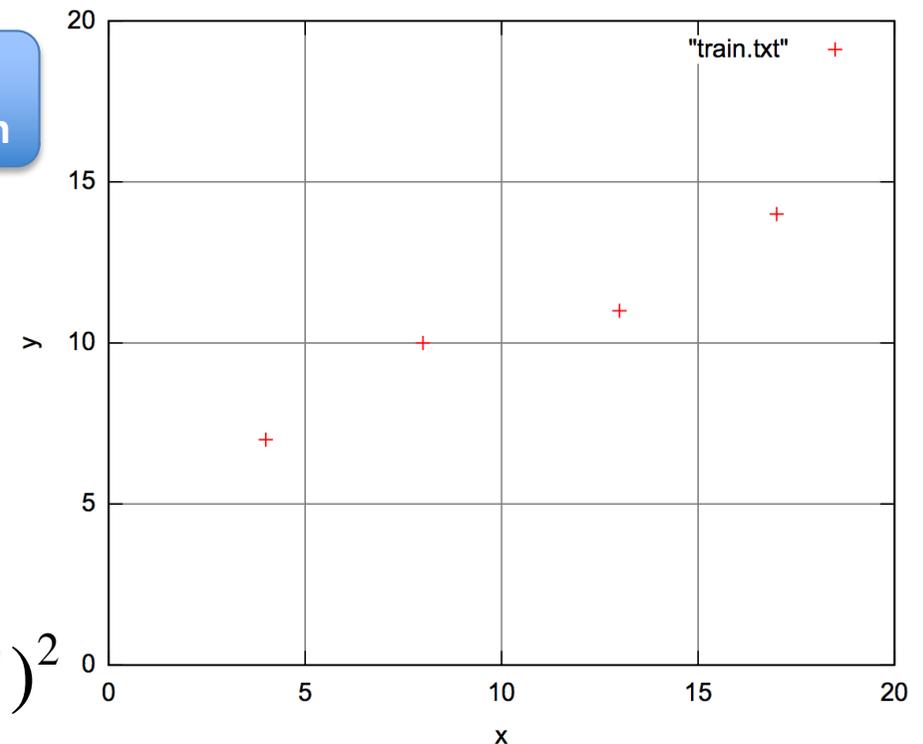
- **Cost function**

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)

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

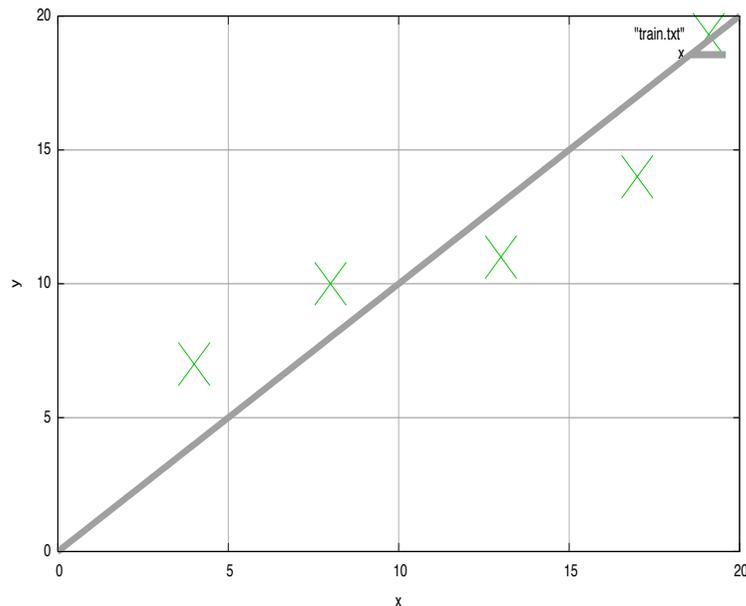


Hypothesis vs. Cost function ($\theta_1=1$)

$\theta_0=0, \theta_1=1, (x,y)=(4,7), (8,10), (13,11)$

Hypothesis:

$$h(x) = x$$

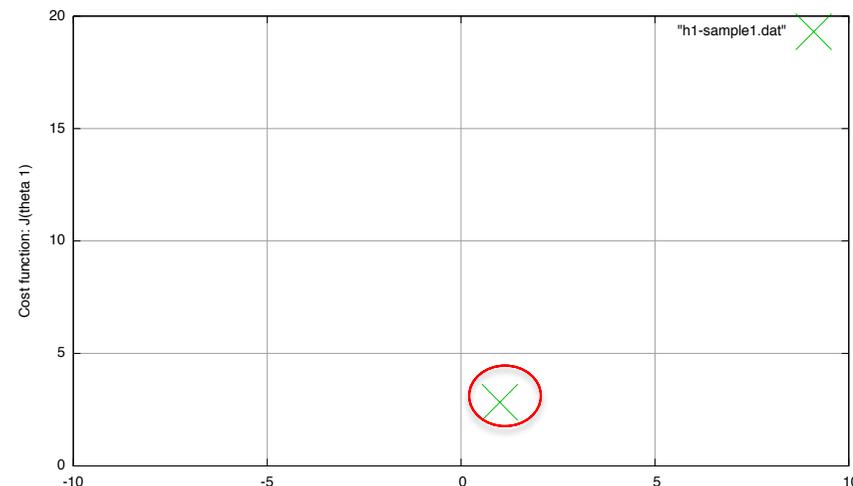


Cost function:

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

$$J(0,1) = \frac{1}{2m} ((4-7)^2 + (8-10)^2 + (13-11)^2)$$

$$J(0,1) = \frac{1}{2 * 3} (9 + 4 + 4) = \frac{17}{6} = 2.83$$

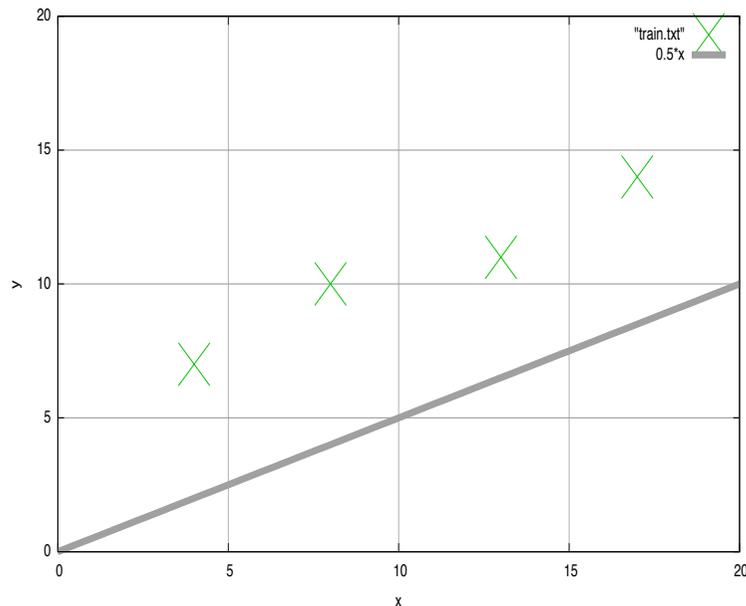


Hypothesis vs. Cost function ($\theta_1=0.5$)

$\theta_0=0, \theta_1=0.5, (x,y)=(4,7), (8,10), (13,11)$

Hypothesis:

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

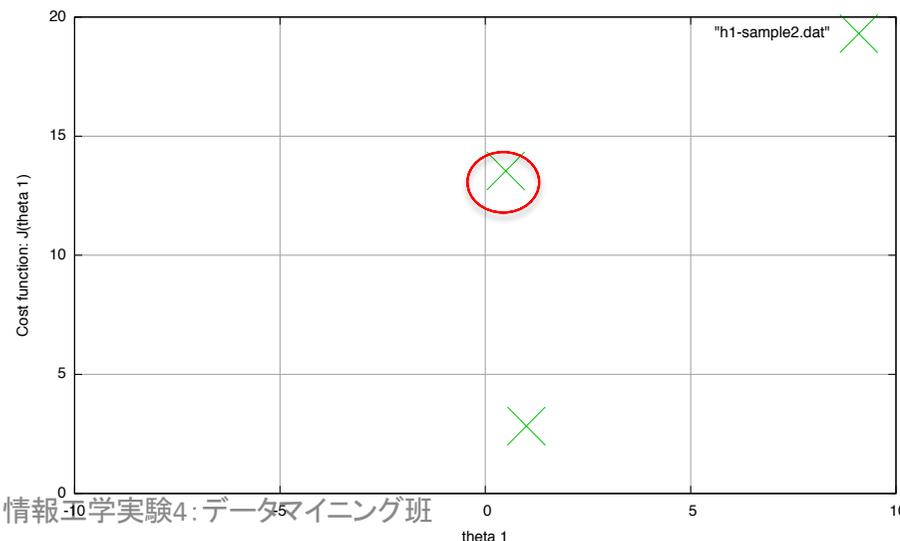


Cost function:

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

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

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



Hypothesis vs. Cost function ($\theta_1 = \text{others}$)

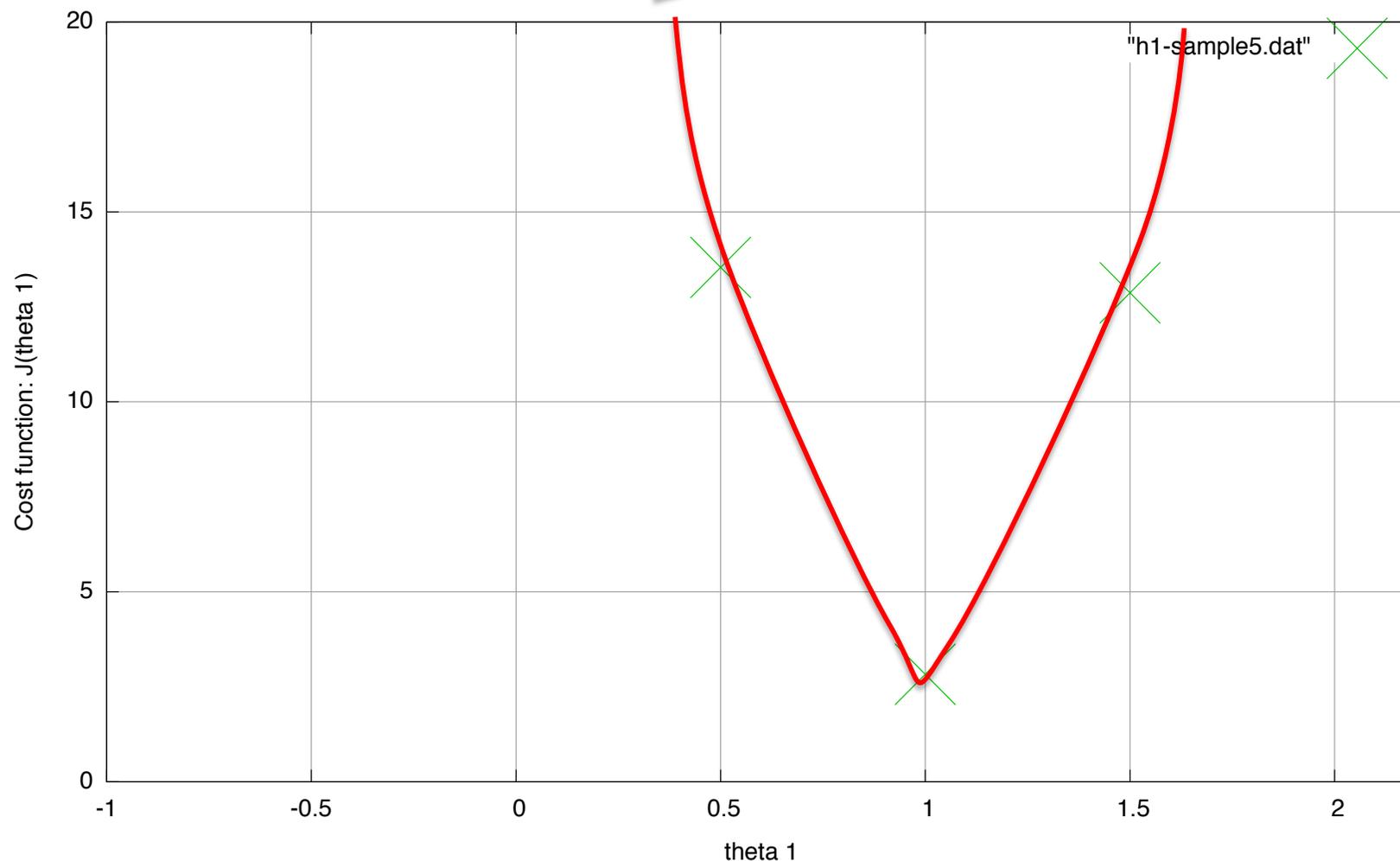
$\theta_0 = 0, \theta_1 = \text{others}, (x, y) = (4, 7), (8, 10), (13, 11)$

- $\theta_1 = 0$:
 - $H(x) = 0 * x = 0$
 - $J(0) = 1/6 \{(0-4)^2 + (0-10)^2 + (0-13)^2\}$
 - $= 1/6 \{16 + 100 + 169\} = 47.5$
- $\theta_1 = 2$:
 - $H(x) = 2 * x$
 - $J(2) = 1/6 \{(8-7)^2 + (16-10)^2 + (26-11)^2\}$
 - $= 1/6 \{1 + 25 + 225\} = 41.83$
- $\theta_1 = 1.5$:
 - $H(x) = 1.5 * x$
 - $J(1.5) = 1/6 \{(6-7)^2 + (12-10)^2 + (19.5-11)^2\}$
 - $= 1/6 \{1 + 4 + 72.25\} = 12.87$

Objective function: minimize $J(\theta_1)$

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

Convex function



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

Ordinary Least Squares

problems?

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

$$7 = \theta_0 + 4\theta_1$$

$$0 = \theta_0 + 4\theta_1 - 7$$

$$e_1 := \theta_0 + 4\theta_1 - 7$$

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

$$E = \sum e_i^2 \geq 0$$

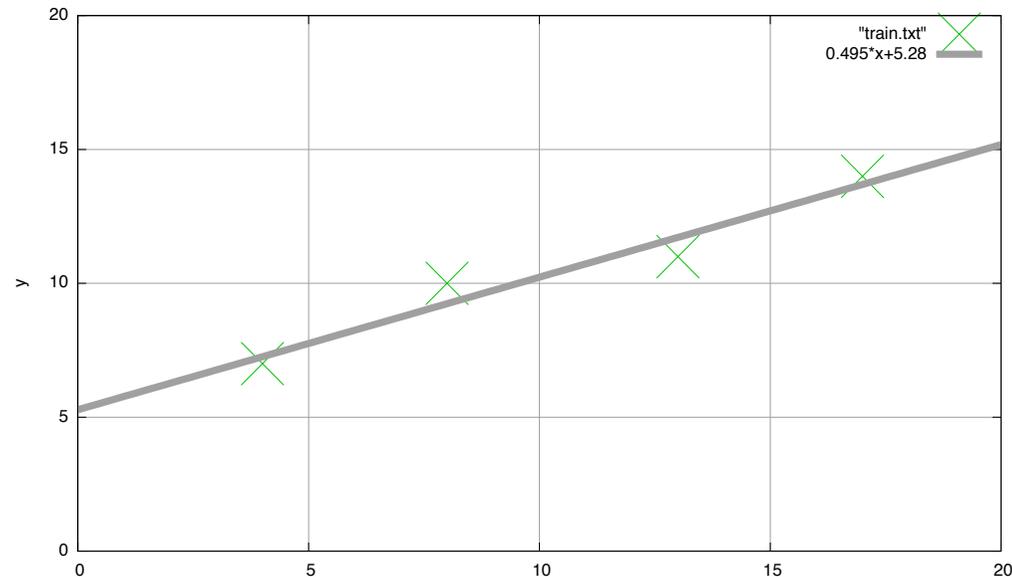
$$= (\theta_0 + 4\theta_1 - 7)^2 + (\theta_0 + 8\theta_1 - 10)^2 + (\theta_0 + 13\theta_1 - 11)^2 + (\theta_0 + 17\theta_1 - 14)^2$$

$$= 538\theta_1^2 + 84\theta_0\theta_1 + 4\theta_0^2 - 978\theta_1 - 84\theta_0 + 466$$

$$= (2\theta_1 + 21\theta_0 - 21)^2 + 97(\theta_0 - 48/97)^2 + 121/97$$

$$\theta_0 = 1029/194 \doteq 5.28, \quad \theta_1 = 48/97 \doteq 0.495$$

$$h(x) = 5.28 + 0.495x$$



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

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

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.

Simple example

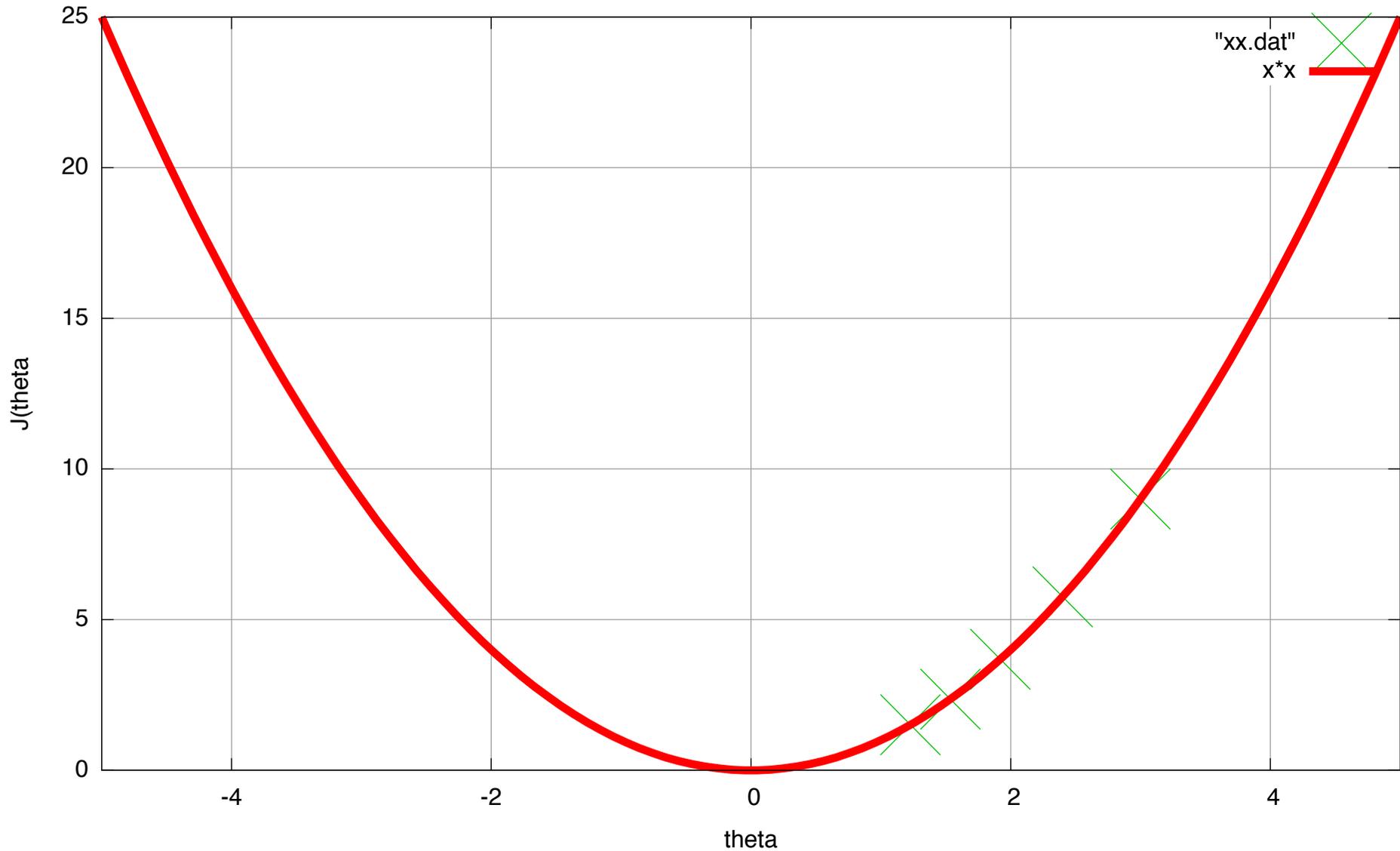
$$J(\theta) = \theta^2, \alpha = 0.1$$

$$\text{new_}\theta = \theta - \alpha \frac{d}{d\theta} J(\theta)$$

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

- e.g., $\alpha=0.1$, $\theta=3$, $J(\theta)=9$
- 1st update
 - $\text{New_}\theta = 0.8 * 3 = 2.4$
 - $J(\theta) = 2.4 ** 2 = 5.76$
- 2nd update
 - $\text{New_}\theta = 0.8 * 2.4 = 1.92$
 - $J(\theta) = 1.92 ** 2 = 3.6864$
- 3rd update
 - $\text{New_}\theta = 1.536$
 - $J(\theta) = 2.359296$
- 4th update
 - $\text{New_}\theta = 1.2288000000000001$
 - $J(\theta) = 1.5099494400000002$

Cont.) the behavior of GD



Gradient descent for Linear Regression

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

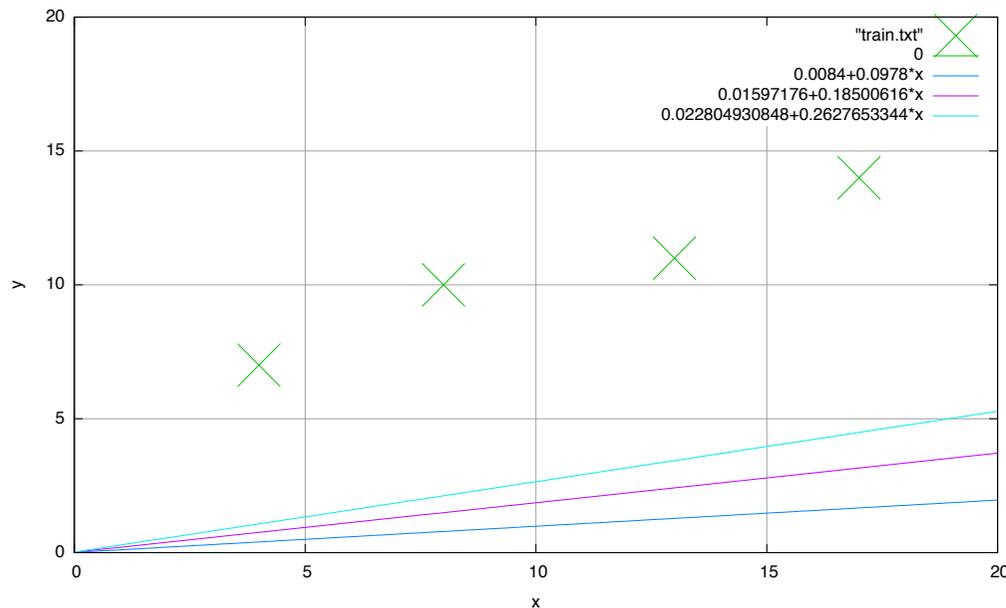
$$\theta_i := \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta_0, \theta_1)$$

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

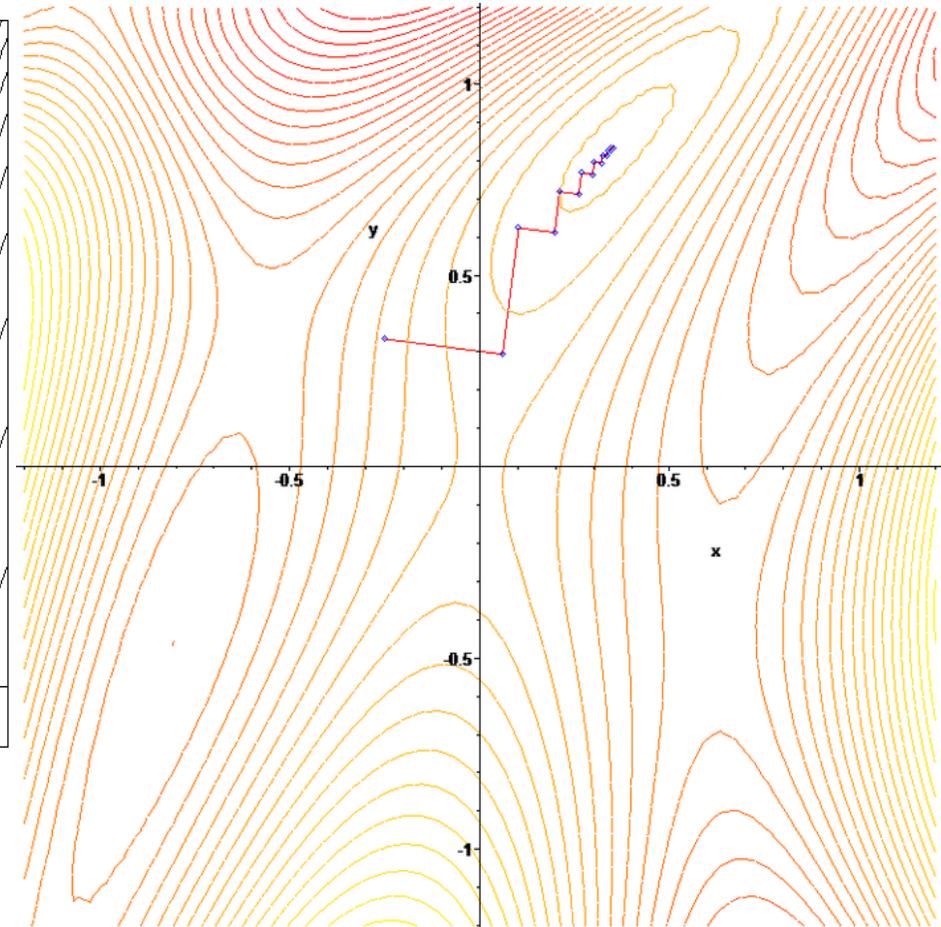
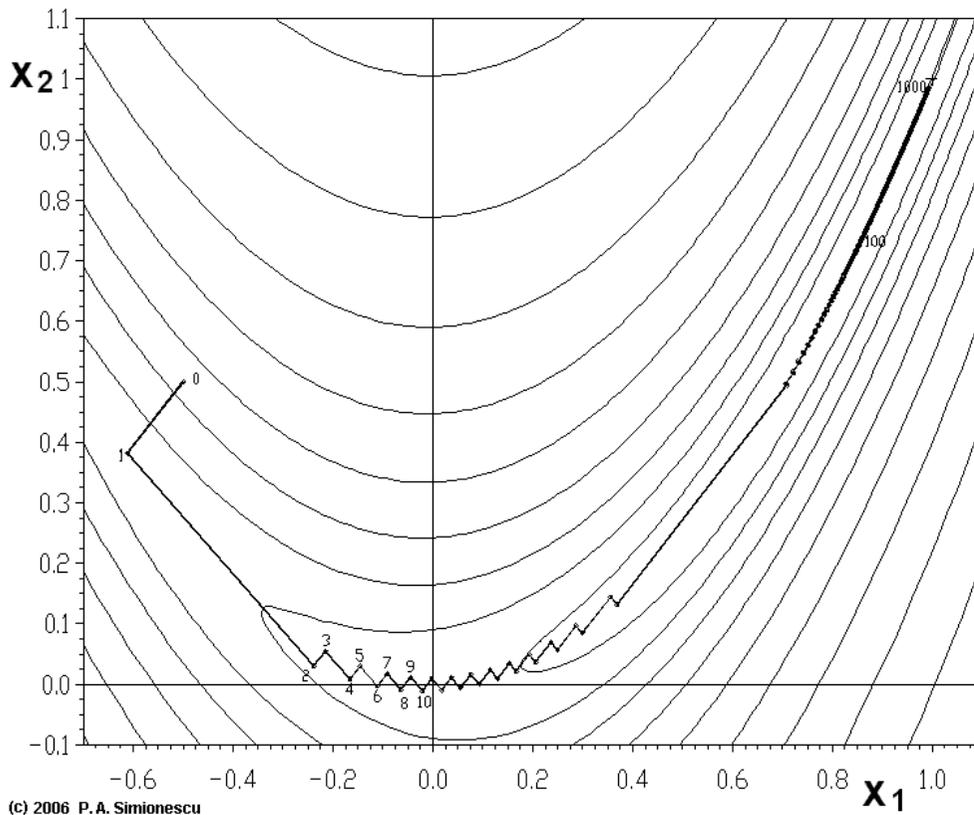
$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = 84\theta_1 + 8\theta_0 - 84$$

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

- e.g., $\alpha=0.001$, $\theta_0=0$, $\theta_1=0$, $J(\theta)=466$
- 1st update
 - $\text{New_}\theta_0 = 0 - 0.001*(-84) = 0.0084$
 - $\text{New_}\theta_1 = 0 - 0.001*(-978) = 0.0978$
 - $J(\theta) = 374.86117384$
- 2nd update
 - $\text{New_}\theta_0 = 0.01597176$
 - $\text{New_}\theta_1 = 0.18500616$
 - $J(\theta) = 302.3858537133122$
- 3rd update
 - $\text{New_}\theta_0 = 0.022804930848$
 - $\text{New_}\theta_1 = 0.2627653344$
 - $J(\theta) = 244.75187010633334$
- 4th update
 - $\text{New_}\theta_0 = 0.0289794580943616$
 - $\text{New_}\theta_1 = 0.3321002229994368$
 - $J(\theta) = 198.91981002677187$



(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>
- An introduction to machine learning with scikit-learn, <http://scikit-learn.org/stable/tutorial/basic/tutorial.html>