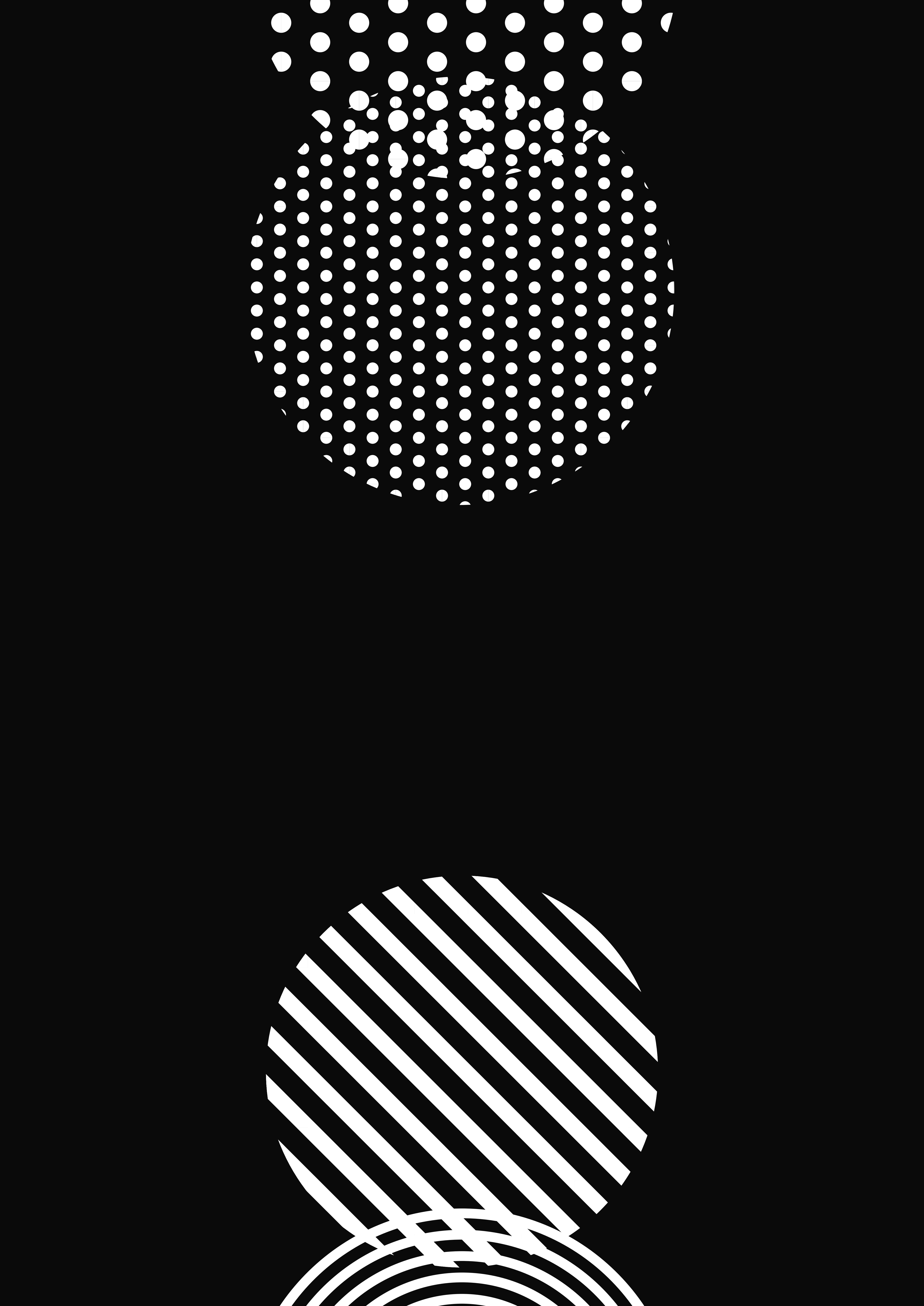


~WEEK 2

今までの線形モデルだと、変数一つにしか対応することができない。

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

特徴の入力をXとしていたが

$X_1\ X_2\ X_3\ X_4\ \dots$ とする。

$X_2^{(3)}$ ⇐ 3番目のテーブル

2個目の特徴

仮説式を複数対応すると、

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$

$\dots + \theta_n \theta x$ となる。

$x_0 = 1$とすると、θとxがベクトルだと考える事ができる。

これを 重回帰（多変量の線形回帰）と呼ぶ。

最急降下法は

$$\theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_1^{(i)}$$

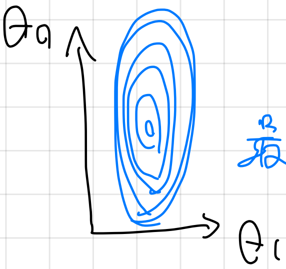
$$\theta_2 := \theta_2 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_2^{(i)}$$

となる。

増えただけ？

Scaling Feature

あまりにパラメータに差があると、等高線図が乱れ、大局的最小値を見つけるのは困難になる。



縦長になる。

最小値は見つけずらい。

$-100 \leq \theta_0 \leq 100 \qquad 0 \leq \theta_1 \leq 1$

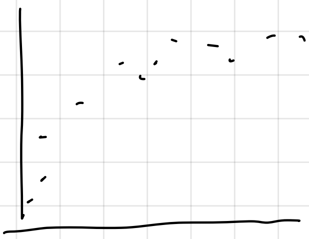
$\theta_a / 100$ などで、パラメータをスケールすると、計算しやすい。 だいたい

$$-1 \leq x_i \leq 1$$

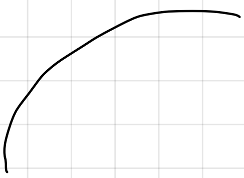
におさえるとよい。

$38 \leq x \leq 50$ のときは？

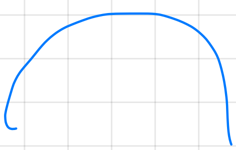
$$\frac{x-38}{20}$$ とするとよい。



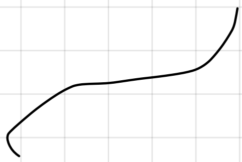
こんなかんじのデータセットに上手く合う線形モデルは？



二次関数？



下降していってない。



三次関数?

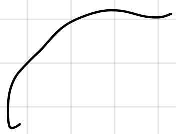
合っているとはいえない。

パラメータに重みをつける。

$x_1 = (\text{パラメータ})^1$

$x_2 = (\text{パラメータ})^2$

$x_3 = (\text{パラメータ})^3$

とすることで、  とできる。

※ この方法を行った時は特にスケーリングに気をつける必要がある。

正規方程式

$$\theta = (X^T X)^{-1} X^T y \quad \cdots ①$$

$X' = X^T$

octave : pinv(X'*X)*X'*y

⇓

一発で大局的最小値が出る。

正規方程式ではスケーリングはしなくてよい。

最急降下法と正規方程式どちらを使うべきか。

- αを指定する必要がある。
- 反復回数が多い。
- 特徴値 n の数が多い時に上手く機能する。

- αを指定する必要がない。
- 反復は必要ない。
- 計算が必要 ⇒ ①
- n が多いと上手く機能しない。

$8836 - 4761$

$= 4075$

$$\begin{matrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{matrix} \qquad \begin{matrix} 1 & 1 \\ 2 & 2 \end{matrix}$$

$$\begin{pmatrix} 5 & 5 \\ 11 & 11 \\ 17 & 17 \end{pmatrix}$$

octaveにおける線形回帰の仮説

- ベクトル化せずに全てのθとxをかける

```
prediction = 0.0;
for j = 1 : n+1,
    prediction = prediction +
                 theta(j) * x(j)
end;
```

- ベクトル化してライブラリを利用

```
prediction = theta' * x;
```

octaveでは転置の意味

$theta' = theta^T$

わかったこと

$\theta^T x$ は二つのベクトル値

（θはベクトルじゃないときもある）

$$
\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}
\qquad
\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}
$$

仮定式は $h_\theta(x) = \theta_0 + \theta_1 x \ldots$

なので

$$
\theta^T = \begin{bmatrix} \theta_0 & \theta_1 & \theta_2 & \theta_3 \end{bmatrix} * \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}
$$

この場合はただの行列の積

octaveでは、

```
sum(v.*w);
```

とすることで、

$$\begin{bmatrix}v_0\\v_1\\v_2\\v_3\end{bmatrix} \quad \begin{bmatrix}w_0\\w_1\\w_2\\w_3\end{bmatrix}$$

.* は同じ位置同士で積

同じことができる。

プログラミング課題1

攻略メモ

sum()?

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

"

$$J\theta(x) = h_\theta(x^{(i)}) + \sim\sim\sim \ ?$$

computeCost において、最急降下法を用いていないのにコスト関数が収束した。なぜ?

→ θ_0 と θ_1 を0に固定しているから答えではない…?

gradient.m は iteration の数だけ呼びだされているっぽい?

◦ 偏微分をプログラムで記述する方法は?

Gradient descent algorithm

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)$$

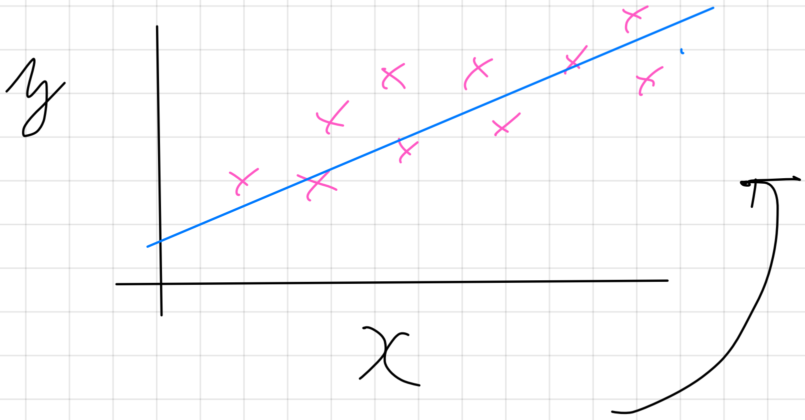
$$\theta_1 := \theta_1 - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)\cdot x^{(i)}$$

}

$$\frac{\partial}{\partial\theta_1}\frac{1}{2m}\sum_{i=1}^{m}(\theta_0+\theta_1 x^{(i)}-y^{(i)})^2$$

x が y となるデータに

青線（線形モデル）をひきたい。



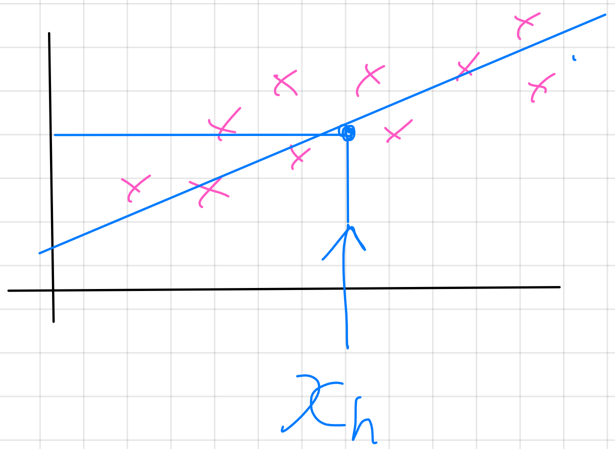
こんなかんじの線形モデルがほしい。

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

なんでほしいの？

線形モデルが得られると、

x から y を推定できる。

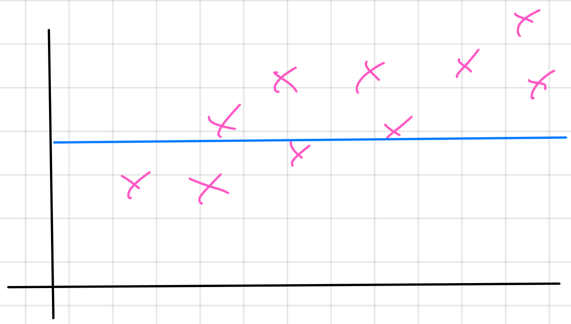


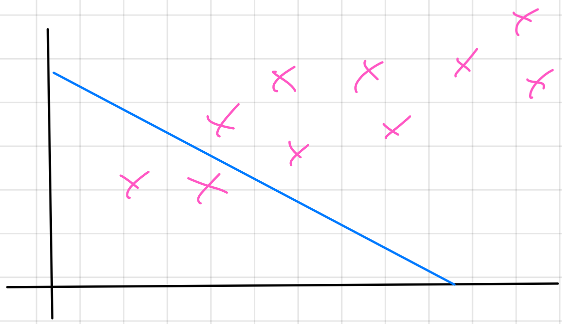
コノヘン カナ…？

どうやって線を引かせる？

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

において、 （Hypothesis）





機械的に正しい $h_\theta(x)$ を得るのは難しい。（何を基準にして線形モデルを描くかを定める）

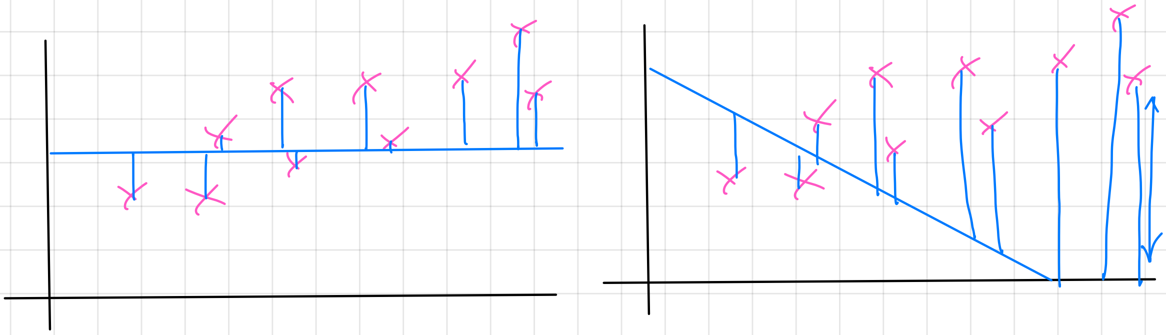
そこで、コスト関数を使う。

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

- mはデータのテーブル数（データ総数）
- Hypothesis － 実際のデータ（誤差量）
- 最小二乗法とかいうやつ

（なんで二乗するのかわからん…）

コスト関数で、どれくらいまちがった線を引いてるのかはわかった。

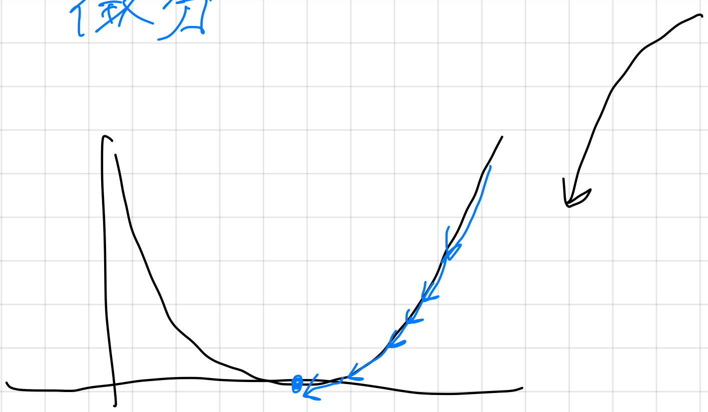


$h_\theta(x)$ と × の距離が一番近い（小さい）状態になる線形モデルを探す。

その方法として、最急降下法を使う。
（Gradient Descent）

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

- ラーニングレート
- 微分



適当な値から計算を始めて、前の地点から傾きを引く。

正規方程式 (Normal equation)

$$\theta = (X^T X)^{-1} X^T y$$

上記の式で数学的にθを一発で求めることができる。

多項式回帰 (重回帰)

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 \sqrt{x}$$

変数が一種類 (xのみ) でも、