2018年6月4日更新

Exercises in Computer-Aided Problem Solving

5. 最小二乗法と線形近似

Least-square method & line fitting

東北大学 大学院工学研究科

嶋田 慶太　shimada@m.tohoku.ac.jp

この副資料の場所:

http://www.pm.mech.tohoku.ac.jp/member/class

TOHOKU UNIVERSITY

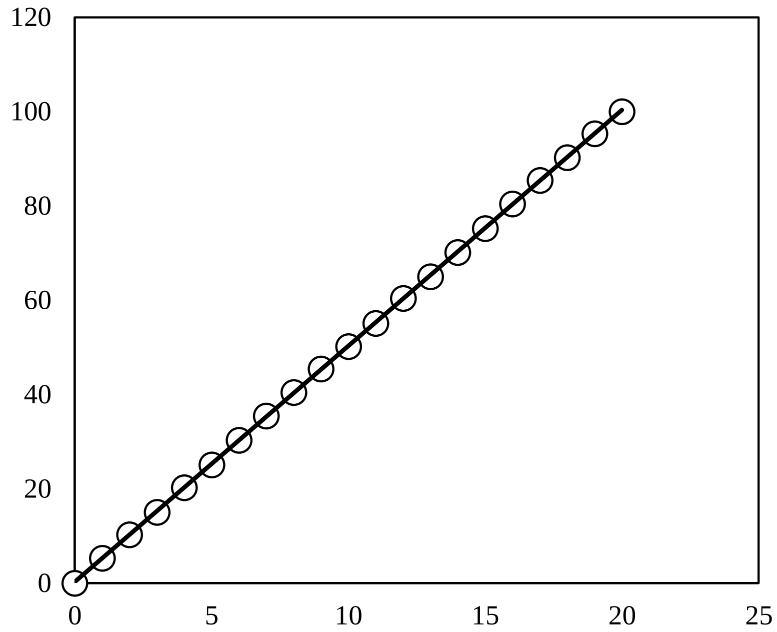
本日の内容

- データの整理
- 列ベクトルと行列の表現
- モデル関数と近似
- 最小二乗法による線形近似

教科書は内容が省かれているので,
前段階の説明をします.

入力・出力の情報を整理する

入力	出力
0	0
1	5.24
2	10.15
3	15.13
4	20.5
5	25.43
6	30.31
7	35.01
8	40.23
9	45.22
10	50.22
11	55.27
12	60.37
13	65.3
14	70
15	75.07
16	80.18
17	85.38
18	90.42
19	95.42
20	100.41



$$y = 5x$$

といえると情報が整理できる

この時にデータ（入力・出力）
に基づいて数式を得る方法を考える

この授業での行列・列ベクトルの表現

列ベクトル

▶ 波括弧{}で表現

*世界的にこれがスタンダードというわけではないので
もし別の機会で使おうと思ったら，ルールを書いてください．

$$\begin{Bmatrix} x_1 \\ \vdots \\ x_n \end{Bmatrix} \xrightarrow{\text{縮記}} \{x_i\}_{i=[1,\,n]} \xrightarrow{\text{縮記}} \{x_i\}$$

誤解がない場合は

例えば
$\{1\}$　$\{0\}$
は全要素が
1，0の列ベクトル

* $\boldsymbol{x}$のように太字もよく使う．
* 行ベクトルは$\{x_i\}^T$のように列ベクトルの転置で表す

行列

▶ 大文字もしくは角括弧[]で表現

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$$

例えば

$$[\{x_i\}\{y_i\}]$$

は i 行の成分がx_i, y_iの $n\times 2$ 行列

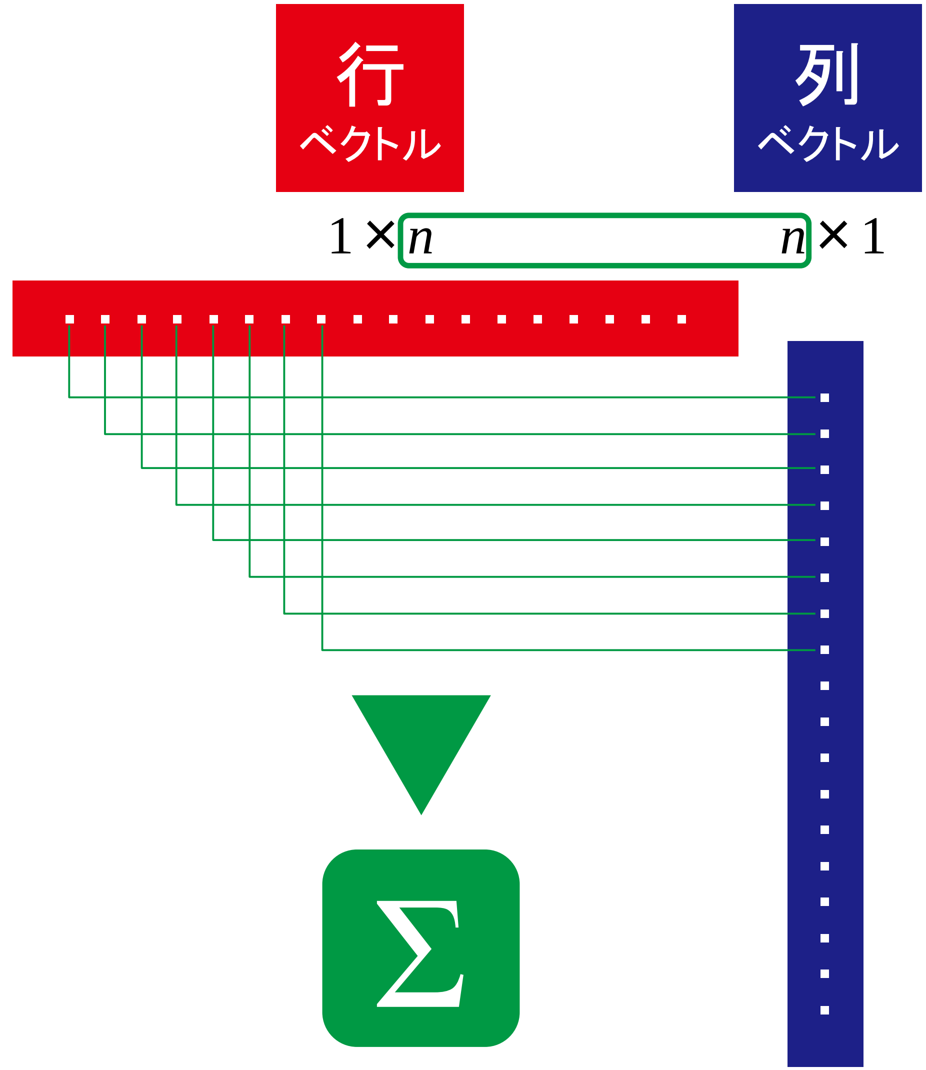
行列・列ベクトルによる総和の書き方

$$\sum_{i=1}^{n} 1 = \{1\}^T\{1\} = n$$

$$\sum_{i=1}^{n} x_i = \{1\}^T\{x_i\} = \{x_i\}^T\{1\}$$

$$\sum_{i=1}^{n} x_i^2 = \{x_i\}^T\{x_i\}$$

$$\sum_{i=1}^{n} x_i y_i = \{y_i\}^T\{x_i\} = \{x_i\}^T\{y_i\}$$



簡単に書けるでしょ！？

計算の検証


```
N = 50;
x = rand(N,1);
y = rand(N,1);
rows(x)
sum(ones(size(x)))
ones(size(x))'*ones(size(x))
sum(x)
ones(size(x))'*x
sum(x.^2)
x'*x
sum(x.*y)
y'*x
```

実験データ：入力と出力の関係

一般的な実験の目的　入力 x に対する出力 y を調べ，関係を見つける

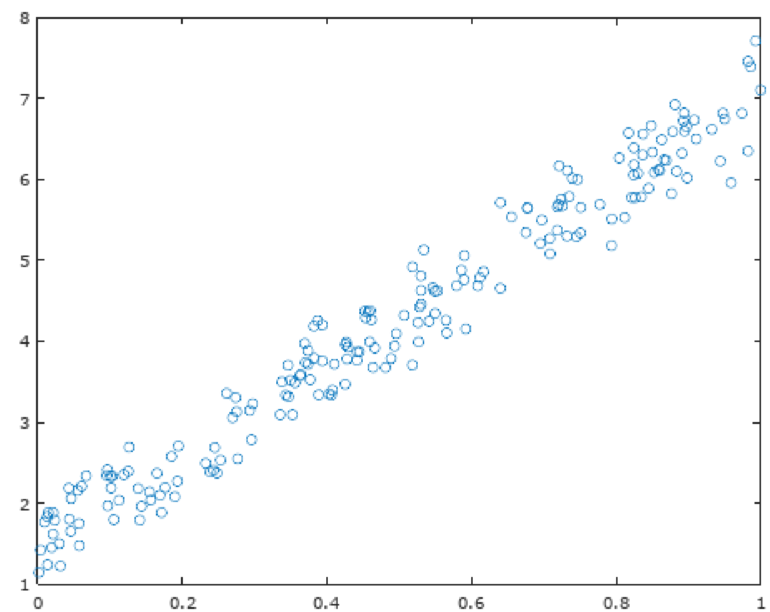
例えば，
入力 ▸ 電流I
出力 ▸ 電圧V
オームの法則　$V = RI$
が見出されたように.

入力 $\{x_i\}$ → 真の関数 f_t → **出力** $\{y_i\}$

実測値　　実測値

入力・出力の対応を
真の関数が記述するはず

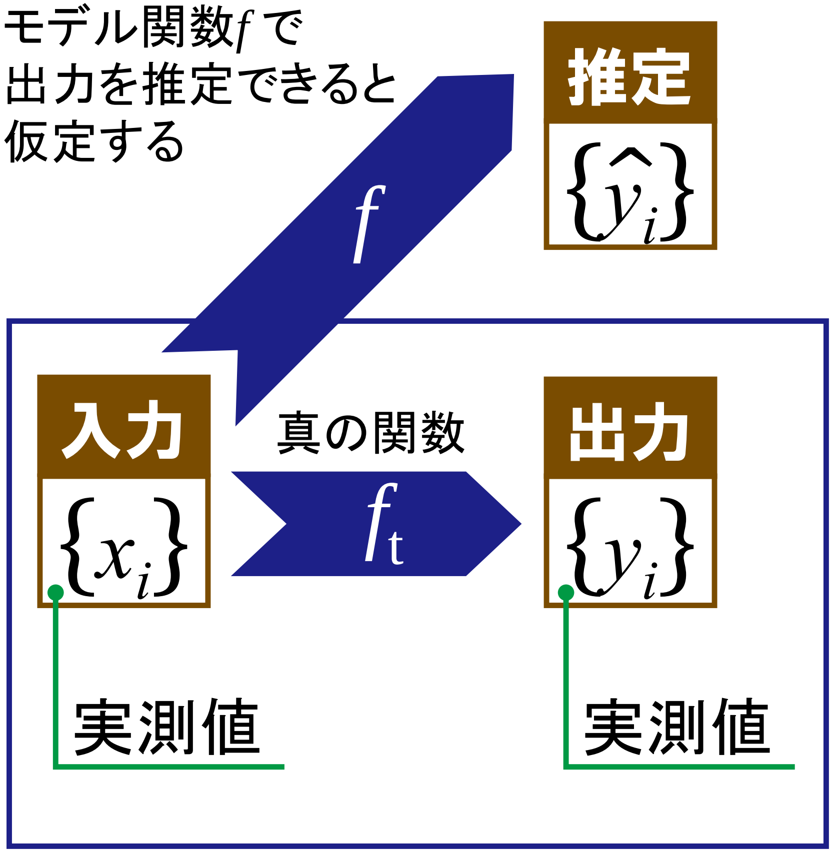
しかし



現実には種々の要因によりばらつく

モデル関数：入力と出力を関数で近似

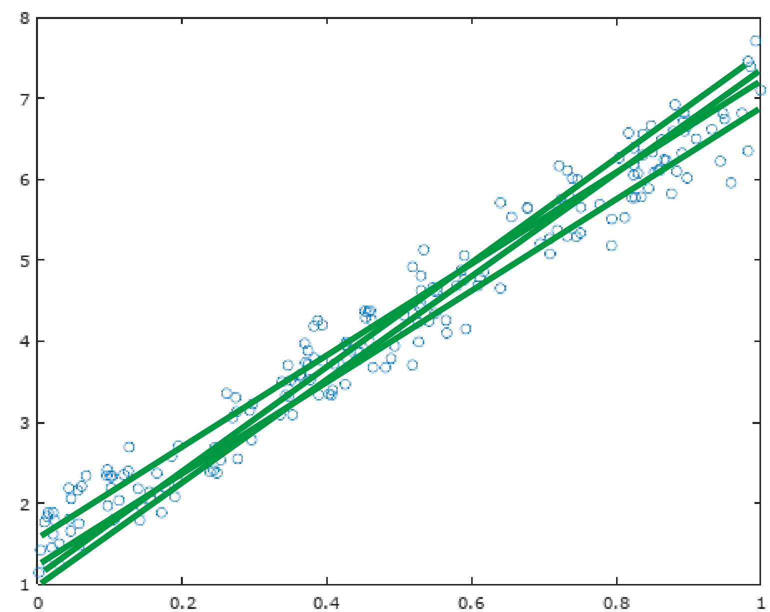
モデル関数fで
出力を推定できると
仮定する



例えばモデル関数を一次関数とすると，

?

それらしい線は
沢山引けてしまう



残差 $\varepsilon_i = y_i - \hat{y}_i$ の総和を最小にする．ただし…

最小二乗法による1次関数へのフィッティング

$$E = \sum_{i=1}^{n} \left(y_i - (ax_i + b)\right)^2$$

残差ベクトル

$$\{\varepsilon_i\} = \{y_i\} - [\{x_i\}\{1\}] \begin{Bmatrix} a \\ b \end{Bmatrix}$$

$$= \{\varepsilon_i\}^T \{\varepsilon_i\}$$

スライド5 の 変形

!

$\{x_i\}$, $\{y_i\}$: 既知値（実験から得られていると考えて）

a, b: 未知値 ◀ 変化させて E を最小にする

偏微分して極値を出す

$$\frac{\partial E}{\partial a} = 0, \quad \frac{\partial E}{\partial b} = 0$$

係数についての偏微分

条件 (1)

$$\frac{\partial E}{\partial a} = \frac{\partial}{\partial a}(\{\varepsilon_i\}^T\{\varepsilon_i\}) = \left(\overbrace{\frac{\partial}{\partial a}\{\varepsilon_i\}^T}^{-\{x_i\}^T}\right)\{\varepsilon_i\} + \{\varepsilon_i\}^T\left(\overbrace{\frac{\partial}{\partial a}\{\varepsilon_i\}}^{-\{x_i\}}\right) = 0$$

▸ Eの定義により分解

▸ ライプニッツルールにより分解

$$\{x_i\}^T\{\varepsilon_i\} = 0$$

残差ベクトルとその微分

$$\{\varepsilon_i\} = \{y_i\} - [\{x_i\}\{1\}]\begin{Bmatrix} a \\ b \end{Bmatrix}$$

$$\frac{\partial}{\partial a}\{\varepsilon_i\} = -[\{x_i\}\{1\}]\begin{Bmatrix} 1 \\ 0 \end{Bmatrix} = -\{x_i\}$$

$$\frac{\partial}{\partial b}\{\varepsilon_i\} = -[\{x_i\}\{1\}]\begin{Bmatrix} 0 \\ 1 \end{Bmatrix} = -\{1\}$$

条件 (2)

$$\frac{\partial E}{\partial b} = 0$$

$$\{1\}^T\{\varepsilon_i\} = 0$$

(1) (2) を 併せて

$$[\{x_i\}\{1\}]^T\{\varepsilon_i\} = 0$$

まとめると

得られた条件

$$\underbrace{[\{x_i\}\{1\}]}_{X}{}^T\{\varepsilon_i\} = 0$$

X と置いて

残差ベクトルの定義

$$\{\varepsilon_i\} = \{y_i\} - \underbrace{[\{x_i\}\{1\}]}_{X}\begin{Bmatrix} a \\ b \end{Bmatrix}$$

X と置いて

代入

$$\boxed{X^T\left(\{y_i\} - X\begin{Bmatrix} a \\ b \end{Bmatrix}\right) = \{0\}}$$

変形して

$$\boxed{\begin{Bmatrix} a \\ b \end{Bmatrix} = \underline{(X^TX)^{-1}X^T}\{y_i\}}$$

▲ 教科書の擬似逆行列 X^+

pinv(x.^[1,0]) もしくは

pinv([x,ones(size(x))])

最小二乗法のまとめ

最小二乗法は

$\{x_i\}$ 実際の入力 と $\{y_i\}$ 実際の出力 の関係を $f(x)$ モデル関数 で近似する方法.

$\{\hat{y}_i\} = \{f(x_i)\}$ 推定値 と $\{y_i\}$ 実際の出力 の差 $\{\varepsilon_i\} = \{y_i\} - \{\hat{y}_i\}$ 残差ベクトル

の二乗総和 $E = \{\varepsilon_i\}^T\{\varepsilon_i\}$ 残差 が最小になるように f を決める.

線形近似であれば,
偏微分を使い, 整理し,

$$\begin{Bmatrix} a \\ b \end{Bmatrix} = (X^T X)^{-1} X^T \{y_i\}$$

として係数
が得られる

▶ 実は線形近似以外でも使える.

例：他の関数をモデル関数とした場合

それぞれの残差ベクトル $\{\varepsilon_i\}$

(1) 比例の式（$y = ax$）の場合

$$\{\varepsilon_i\} = \{y_i\} - a\{x_i\}$$

(2) 2次関数（$y = ax^2 + bx + c$）の場合

$$\{\varepsilon_i\} = \{y_i\} - \underbrace{[\{x^2{}_i\}\{x_i\}\{1\}]}_{X\text{と置いて}} \begin{Bmatrix} a \\ b \\ c \end{Bmatrix}$$

(3) 正弦関数（$y = a\sin x$）の場合

$$\{\varepsilon_i\} = \{y_i\} - a\underbrace{\{\sin(x_i)\}}_{X\text{と置いて}}$$

スライド9-11の1次関数と同様にたどると，

$$a = (\{x_i\}^T\{x_i\})^{-1}\{x_i\}^T\{y_i\}$$

$$\begin{Bmatrix} a \\ b \\ c \end{Bmatrix} = (X^TX)^{-1}X^T\{y_i\}$$

$$a = (X^TX)^{-1}X^T\{y_i\}$$

一般的な形が見えてきたのでは…？

教科書の例題（スクリプトファイルにすると）

教科書の例はp.48とp.50を順々にコマンドした場合である．それをまとめると…

Years of service	<5	<10	<15	<20	<25	<30	<35
Salary (mil. JPY)	370.8	459.4	533.8	597.7	669.7	719.7	753.8

```
years=[5:5:35]';
income=[371;460;534;598;670;720;754];
plot(years,income,"o")
set(gca,"fontsize",14)
X=years.^[1,0];
p=pinv(X)*income;
hold on
plot([0:1:40],p(1)*[0:1:40]+p(2))
axis([0,40,0,900])
```

◀ データ格納（最初から列ベクトル）

上はプライム記号，
下はセミコロン区切りに気を付ける

```
year=          income=
    5              371
   10              460
   15              534
   20              598
   25              670
   30              720
   35              754
```

教科書の例題(スクリプトファイルにすると)

教科書の例はp.48 と p.50を順々にコマンドした場合である. それをまとめると…

Years of service	<5	<10	<15	<20	<25	<30	<35
Salary (mil. JPY)	370.8	459.4	533.8	597.7	669.7	719.7	753.8

```
years=[5:5:35]';
income=[371;460;534;598;670;720;754];
plot(years,income,"o")
set(gca,"fontsize",14)
X=years.^[1,0];
p=pinv(X)*income;
hold on
plot([0:1:40],p(1)*[0:1:40]+p(2))
axis([0,40,0,900])
```

オリジナルデータからのグラフ作成
`plot`: x軸, y軸, プロットの種類.
第三引数のoはプロットの形
その他は"x"や"@"などいろいろ

`gca`: graphic current axes

教科書の例題(スクリプトファイルにすると)

教科書の例はp.48 と p.50を順々にコマンドした場合である．それをまとめると…

Years of service	<5	<10	<15	<20	<25	<30	<35
Salary (mil. JPY)	370.8	459.4	533.8	597.7	669.7	719.7	753.8

```
years=[5:5:35]';
income=[371;460;534;598;670;720;754];
plot(years,income,"o")
set(gca,"fontsize",14)
X=years.^[1,0];
p=pinv(X)*income;
hold on
plot([0:1:40],p(1)*[0:1:40]+p(2))
axis([0,40,0,900])
```

years(横軸)から係数行列を作成
教科書とは違う方法の例

要素ごと計算なので，

$$X=\begin{array}{cc} \texttt{year.^1} & \texttt{year.^0} \\ \end{array}$$

$$X=\begin{pmatrix} 5 & 1 \\ 10 & 1 \\ 15 & 1 \\ 20 & 1 \\ 25 & 1 \\ 30 & 1 \\ 35 & 1 \end{pmatrix}$$

教科書の例題(スクリプトファイルにすると)

教科書の例はp.48 と p.50を順々にコマンドした場合である. それをまとめると…

Years of service	<5	<10	<15	<20	<25	<30	<35
Salary (mil. JPY)	370.8	459.4	533.8	597.7	669.7	719.7	753.8

```
years=[5:5:35]';
income=[371;460;534;598;670;720;754];
plot(years,income,"o")
set(gca,"fontsize",14)
X=years.^[1,0];
p=pinv(X)*income;
hold on
plot([0:1:40],p(1)*[0:1:40]+p(2))
axis([0,40,0,900])
```

hold on で前のグラフを保持
(これがないと, 次の plot の指示で
前のグラフは消える)

近似直線を上書き.
なお plot の第三引数は省略

x軸とy軸の範囲を指定

教科書の例題(スクリプトファイルにすると)

教科書の例はp.48とp.50を順々にコマンドした場合である．それをまとめると…

Years of service	<5	<10	<15	<20	<25	<30	<35
Salary (mil. JPY)	370.8	459.4	533.8	597.7	669.7	719.7	753.8

別解

```
years=[5:5:35]';
income=[371;460;534;598;670;720;754];
X=years.^[1,0];
p=pinv(X)*income;
plot(years,income,"o",[0:1:40], [0:1:40]'.^[1,0]*p)
set(gca,"fontsize",14)
axis([0,40,0,900])
```

スクリプトファイルの場合，「hold on」せず，また足し算もまとめて
`plot(years,income,"o", [0:1:40],[0:1:40]'.^[1,0]*p)`
としてもよい．

課題　チョコレートとノーベル賞

人口に対するノーベル賞受賞者と
チョコレートの消費量には相関関係がある

	Nobel laureates per capita	Chocolate consumption per capita (kg/y/head)
Sweden	31.855	6.6
Switzerland	31.544	10.8
Denmark	25.255	8.6
Austria	24.332	7.9
Norway	23.368	9.8
UK	18.875	10.3
Ireland	12.706	8.8
Germany	12.668	11.4
USA	10.706	5.1
Hungary	9.038	3.5
France	8.99	7.4
Belgium	8.622	6.8
Finland	7.6	7
Australia	5.451	6
Italy	3.265	3.3
Poland	3.124	4.5
Lithuania	2.836	6.1
Greece	1.857	4.5
Portugal	1.855	4.5
Spain	1.701	3.3
Japan	1.492	2.2
Bulgaria	1.421	2.2
Brazil	0.05	2.5

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3743834/

課題について

課題1

最小二乗法による線形フィッティングを行い，
結果を描画するスクリプトファイルの作成せよ．

必須

補足1: データは授業HPからダウンロードできる．
補足2: データのロードは，作業ディレクトリにファイルを置き

```
>>load "Nobel_vs_choco.txt"
```

で行う．▶ data という変数に格納される．

課題2

最小二乗法により $a_k x^k + a_1 x + a_0$ という形式で
フィッティングを行い，
結果を描画するスクリプトファイルの作成せよ．
ここで k は各自の学籍番号の整数部の1/1000とする．
（すなわちB7TB1357であれば $k = 1.357$）

任意

おまけ　多項式近似に拡張すると

$$E = \sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{m} a_j x^{j-1}\right)^2$$

Octaveでは
x.^[m-1:-1:0]
と容易に表現できる.

要素を $(m$-1)乗　要素を $(m$-2)乗　1乗　0乗

残差ベクトル

$$\{\varepsilon_i\} = \{y_i\} - \left[\{x_i^{m-1}\}\{x_i^{m-2}\}\ldots\{x_i\}\{1\}\right]\{a_j\}$$

n次列ベクトル　m次列ベクトル

$$= \{\varepsilon_i\}^T\{\varepsilon_i\}$$

$$\frac{\partial \mathrm{E}}{\partial a_j} = 0 \ (j = 1 \cdots m)$$

$$\boxed{\{a_j\} = (X^T X)^{-1} X^T \{y_i\}}$$

さらなる一般化は各自調べてみよう.

おまけ　Octaveには多項式近似関数がある

▶ x y を入出力の列ベクトルとして,

1行(n+1)列の計算結果を出力

```
polyfit(x, y, n)
```

ans = ○　○　・・・・　○

x^nの係数　x^{n-1}の係数　定数項

```
pinv(x.^[n:-1:0])*y
```

ans =

○ ◀ x^nの係数

○ ◀ x^{n-1}の係数

・

・

○ ◀ 定数項