横浜国立大学集中講義「現代物理学」
第4回　2021年11月25日

線形回帰と最小二乗法

岡山大学 異分野基礎科学研究所

大槻純也

前回までの復習

ベイズの定理

事後確率
posterior probability

事前確率
prior probability

$$p(\boldsymbol{w}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{w})p(\boldsymbol{w})$$

尤度関数
likelihood function

モデルパラメータ： $\boldsymbol{w}$

観測データ： $\mathcal{D}$

$\boldsymbol{w}$ を知りたい
そのために測定をして$\mathcal{D}$ を得る

$$p(\boldsymbol{w}) \longrightarrow p(\boldsymbol{w}|\mathcal{D})$$

測定により確率が更新される

ここまでの例では、p(D|w)は統計データとして与えられていた
物理で応用する場合は、モデルを仮定してp(D|w)を計算する
(モデル＝関数形、ハミルトニアン)

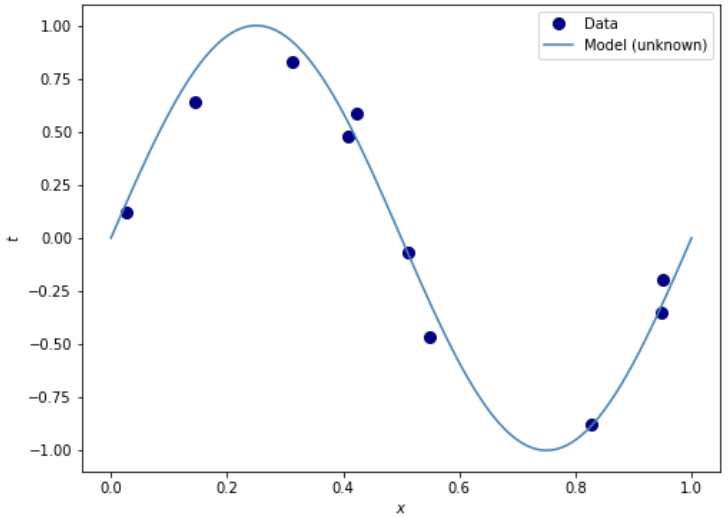
今回からやること

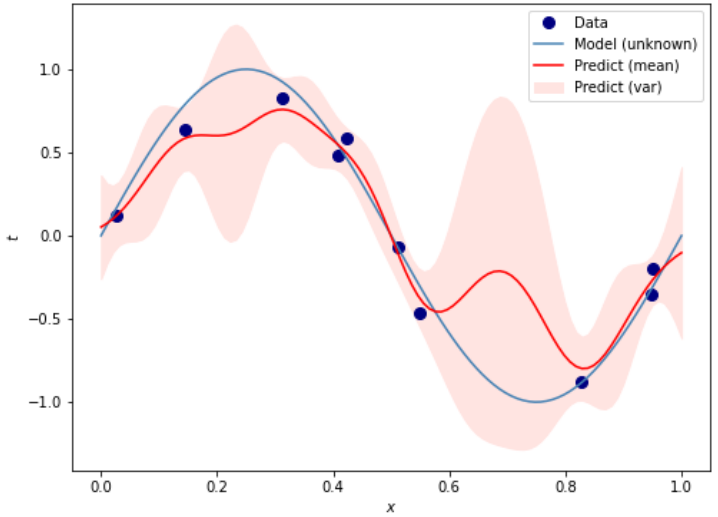
データのフィッティングをしたいとき
→最小二乗法を使う　$t = f(x)$
ただし、関数形が分かっていることが前提

関数形が分からない場合は？
→人間がグラフを見て関数形を推測

多次元の場合は？　$t = f(x_1, x_2, \cdots, x_D)$
→がんばってグラフを描く
or あきらめる

ベイズの定理を応用して　$p(\boldsymbol{w}|\mathcal{D})$
・関数形を決める
・予測する、測定点を決める





用語説明：回帰（Regression）

回帰 (Regression) とは：

データの組 $\{(x_n, t_n)\}$ が与えられたときに、
任意の x に対して t を予測する関数 t = $y(x)$ を見つけること

Training set

$$\mathbf{x} = \{x_1, x_2, \cdots, x_N\}$$

$$\mathbf{t} = \{t_1, t_2, \cdots, t_N\}$$ target

表記はPRMLに従う

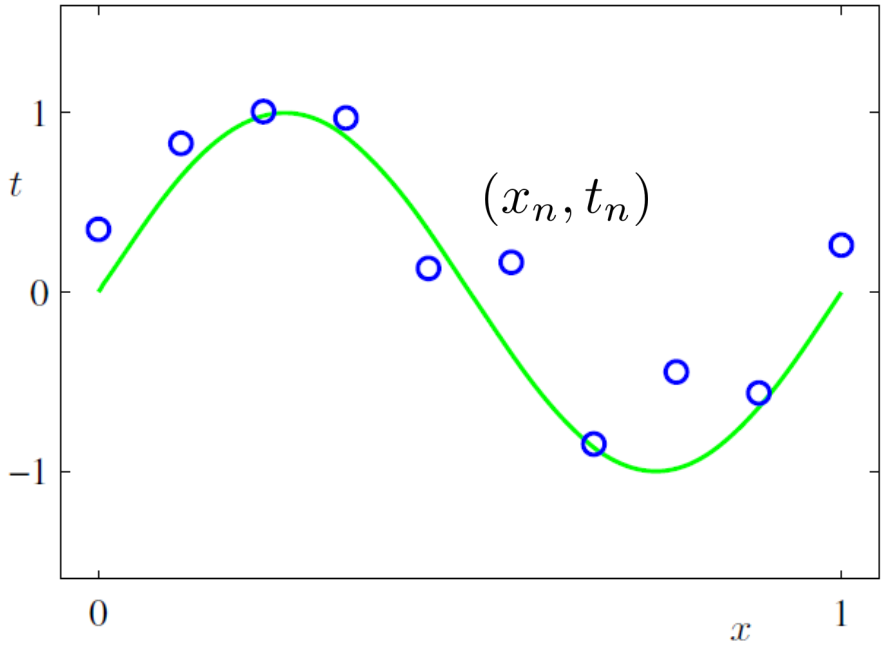
簡単に言えばフィッティング。ただし

- 関数形は未知（関数形も含めて決定する）
- x は多次元 $\boldsymbol{x} = (x_1, x_2, \cdots, x_D)$

機械学習の言葉を使うと

► 教師あり学習
  - t が連続 → 回帰 (Regression)
  - t が不連続 → 分類 (Classification)

► 教師なし学習



PRML Fig. 1.2

例：多項式フィッティング

まずは関数 $y(x)$ として多項式 (polynomial) を考える

$$y(x, \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M$$
$$= \sum_{j=0}^{M} w_j x^j$$

パラメーター $\{w_j\}$ に関して線形 (linear)

➔ 線形回帰 (Linear regression)

入力 x に対しては非線形 (nonlinear)

一般的には

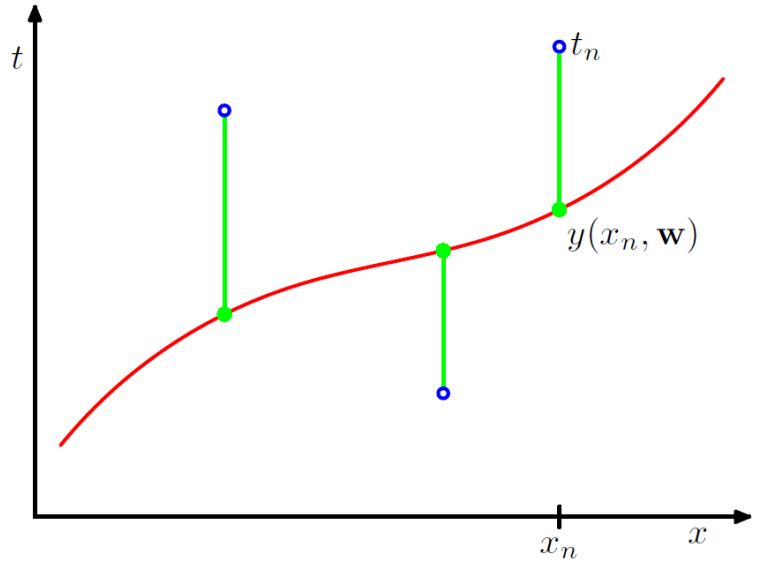
$$y(x, \boldsymbol{w}) = \sum_{j=0}^{M} w_j \phi_j(x)$$

基底関数 (basis function)
非線形

誤差関数 (Error function)

$$E(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} [y(x_n, \boldsymbol{w}) - t_n]^2$$

関数 $y(x, \boldsymbol{w})$ がデータ $\{(x_n, t_n)\}$ をどの程度表現できてるか



PRML Fig. 1.3

最小二乗法 (Least squares)

誤差関数 (Error function)

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\left[y(x_n, \boldsymbol{w}) - t_n\right]^2$$

$$y(x, \boldsymbol{w}) = \sum_{j=0}^{M} w_j x^j$$

誤差関数を最小化

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} E(\boldsymbol{w})$$

関数を最小にする $\boldsymbol{w}$ を返す演算子
(argument of the minimum)

予測曲線

$$t = y(x, \boldsymbol{w}^*)$$

argminを計算すると

$$\boldsymbol{w}^* = (\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}}\mathbf{t} \equiv \Phi^{+}\mathbf{t}$$

$$\mathbf{t} = (t_1, t_2, \cdots, t_N)^{\mathrm{T}}$$ Tは転置

$$\Phi = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^M \\ 1 & x_2 & x_2^2 & \cdots & x_2^M \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^M \end{pmatrix}$$

$N \times M$ 行列

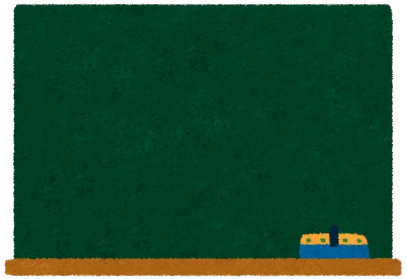
計画行列
design matrix

$$\Phi^{+} = (\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}}$$

ムーア・ペンローズの疑似逆行列
Moore-Penrose pseudo inverse

$\mathbf{t} = \Phi\boldsymbol{w}$ を $\boldsymbol{w}$ について解く
ことに相当するため

ベクトルの微分を使うと便利 (次頁)



公式：ベクトル微分

成分ごとの微分ではなく、ベクトルでの微分を使うと計算が楽になる

定義

$$\frac{\partial}{\partial \boldsymbol{x}} f(\boldsymbol{x}) \equiv \begin{pmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\boldsymbol{x})}{\partial x_N} \end{pmatrix}$$

公式

$$\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}) = \frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}) = \boldsymbol{y}$$

$$\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{x}^{\mathrm{T}}A\boldsymbol{x}) = (A + A^{\mathrm{T}})\boldsymbol{x}$$

Aは($N \times M$)行列

線形回帰では2次形式しか現れないのでこれで十分

誤差関数を行列・ベクトル表記する

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}[y(x_n, \boldsymbol{w}) - t_n]^2$$

$$= \frac{1}{2}(\Phi\boldsymbol{w} - \mathbf{t})^{\mathrm{T}}(\Phi\boldsymbol{w} - \mathbf{t})$$

Tは転置

$$y(x, \boldsymbol{w}) = \sum_{j=0}^{M} w_j x^j = \underline{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(x)} = \underline{\boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{w}(x)}$$

ベクトル同士の内積

ここで

$$\boldsymbol{\phi}(x) = \begin{pmatrix} 1 \\ x \\ \vdots \\ x^M \end{pmatrix}$$

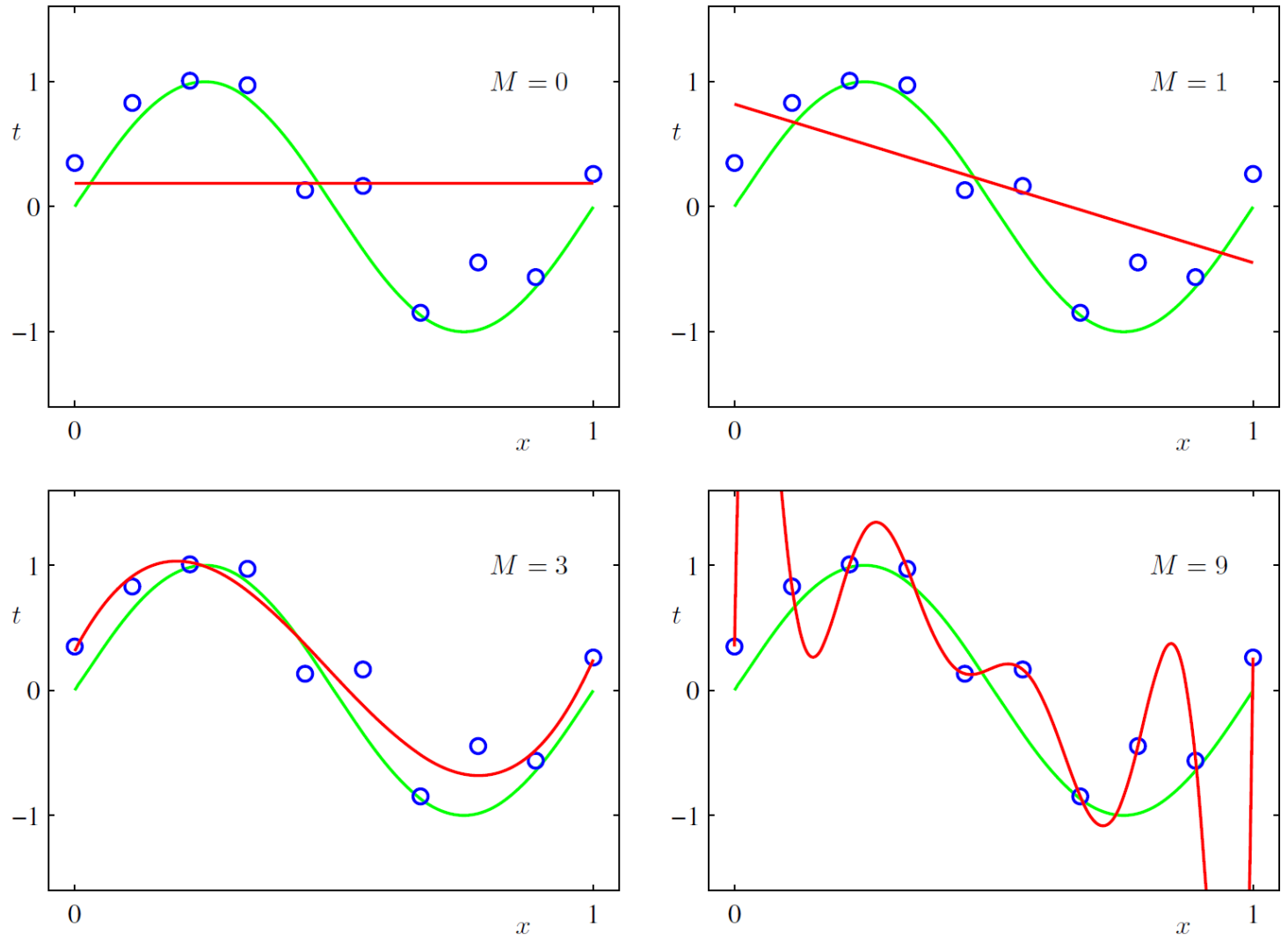
一般的には

$$\boldsymbol{\phi}(x) = \begin{pmatrix} \phi_0(x) \\ \phi_1(x) \\ \vdots \\ \phi_M(x) \end{pmatrix}$$

ベクトルは全て縦ベクトルとして表す

例：多項式フィッティング

$N = 10$
$M = [0:9]$



M=3くらいが妥当

M=9は明らかにおかしい
Overfittingと言う
(過学習、過剰適合)

➔ $E(\boldsymbol{w}^*)$ が小さければ小さいほど良いわけではない

青丸：測定データ
緑：正解
赤：予測曲線

PRML Fig. 1.4

どうやってoverfittingを避けるか

テストデータを利用する

Test set $\mathbf{x}' = \{x'_1, x'_2, \cdots, x'_{N'}\}$
$\mathbf{t}' = \{t'_1, t'_2, \cdots, t'_{N'}\}$

2つの Root-Mean-Square (RMS) error (平均二乗偏差) を定義

Training error

$$E_{\mathrm{RMS}}^{\mathrm{Training}} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}[y(x_n, \boldsymbol{w}^*) - t_n]^2} = \sqrt{\frac{2}{N}E(\boldsymbol{w}^*)}$$

フィッティングの精度を表す

Test error

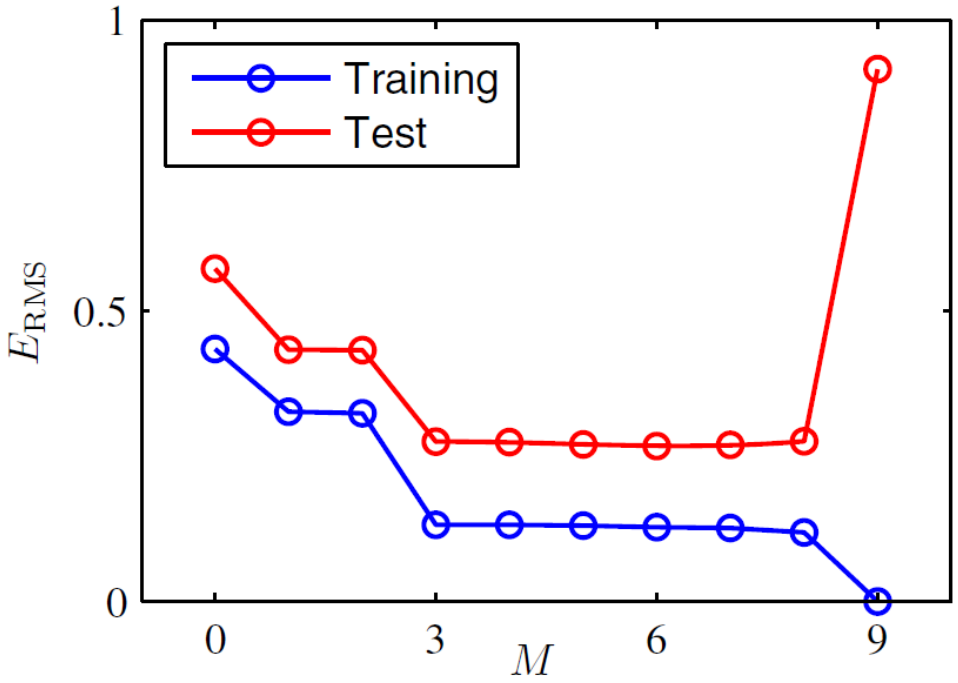
$$E_{\mathrm{RMS}}^{\mathrm{Test}} = \sqrt{\frac{1}{N'}\sum_{n=1}^{N'}[y(x'_n, \boldsymbol{w}^*) - t'_n]^2}$$

汎化性能を表す

問題点：testデータが必要

testデータがない場合には、trainingデータを減らして
testに使う（trainingデータが減ってしまう）。
これを系統的に行う方法を交差検証法と呼ぶ。

generalization (汎化)：
学習に使ったデータとは異なるデータも
正しく予測できること。汎化性能。



PRML Fig. 1.5

もう一つの方法：正則化

正則化 (Regularization)：

誤差関数に項を付け加えて、解を制約すること

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}[y(x_n, \boldsymbol{w}) - t_n]^2 + \frac{\lambda}{2}\sum_j |w_j|^2$$

$$= \frac{1}{2}(\Phi\boldsymbol{w} - \mathbf{t})^{\mathrm{T}}(\Phi\boldsymbol{w} - \mathbf{t}) + \frac{\lambda}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$$

$\boldsymbol{w}$ のノルムにペナルティを課す

λ はhyper parameterと呼ばれる

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

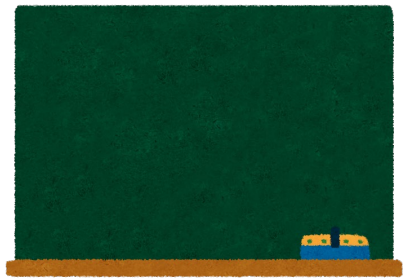
PRML Table 1.1

$\frac{\partial E(\boldsymbol{w})}{\partial \boldsymbol{w}} = 0$ より

$$\boldsymbol{w}^* = (\Phi^{\mathrm{T}}\Phi + \lambda\mathbf{I})^{-1}\Phi^{\mathrm{T}}\mathbf{t}$$

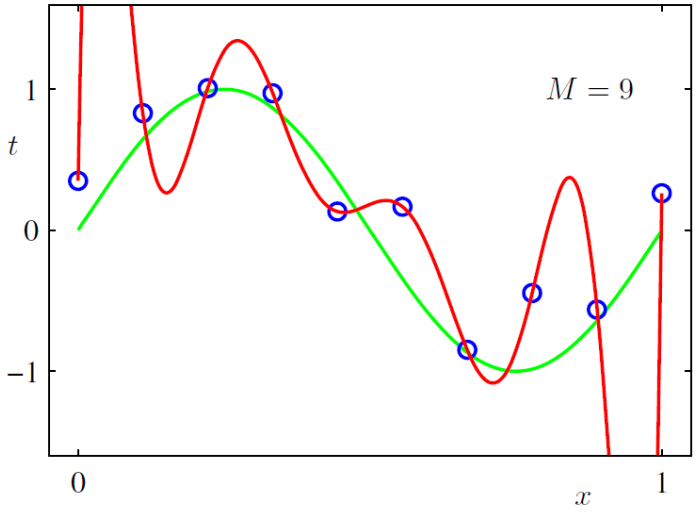
リッジ回帰
Ridge regression

最小二乗解との違い ($\mathbf{I}$ は単位行列)
逆行列計算を安定化させる (ランク落ちを防ぐ)

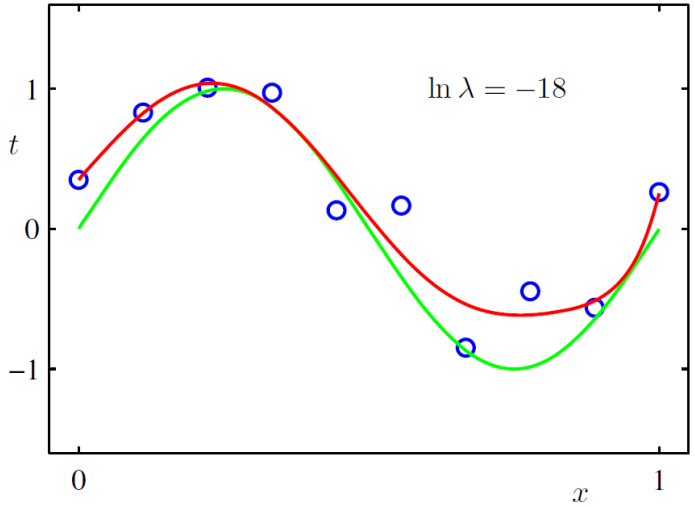


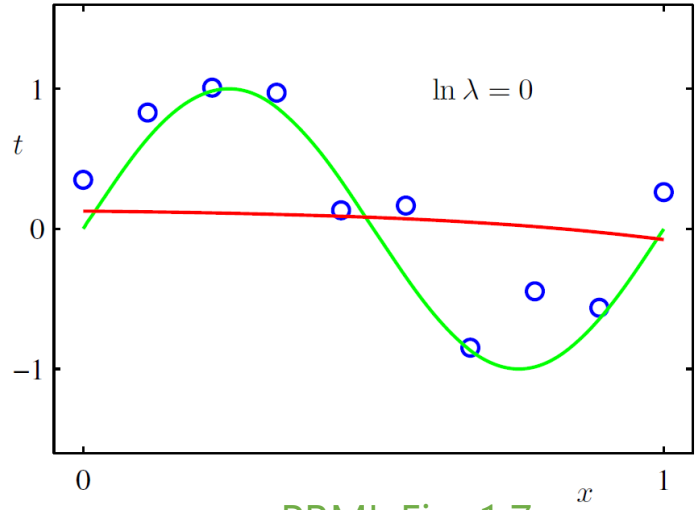
結果

正則化なし
($\ln \lambda = -\infty$)



正則化あり





PRML Fig. 1.7

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

PRML Table 1.2

overfittingが抑えられている

ただし、λ が大きすぎるとデータの特徴が消えてしまう

適切な λ の値をどのように決めるかという問題が残る。とりあえずは、$\lambda = 10^{-8}$ のように非常に小さくとる。