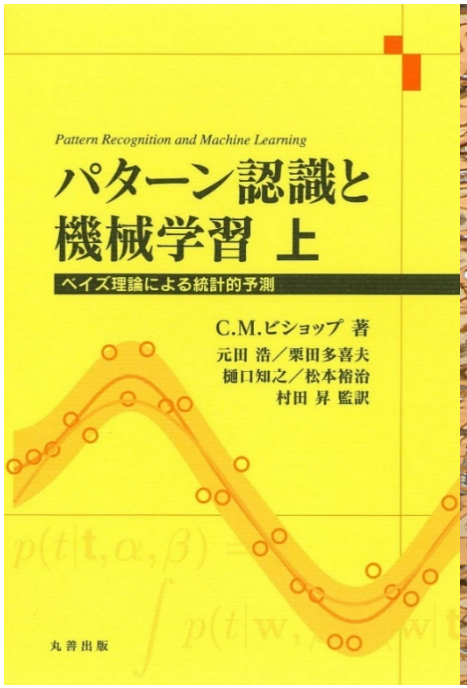
PRMLゼミ

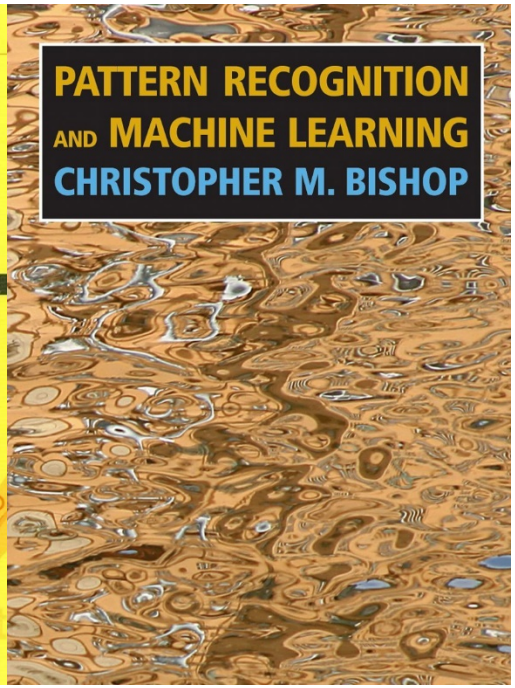
1章イントロ・1.1節・1.3節

anmitsu48

本資料について

- 本資料は、『パターン認識と機械学習　上　－ベイズ理論による統計的予測－』(丸善出版)を用いてゼミを行った際に、私が使用した発表資料を再編集したものである。
- 再編集の際は、私が持っている他の資料も利用した。
  参考にした資料は最後にまとめて紹介する。



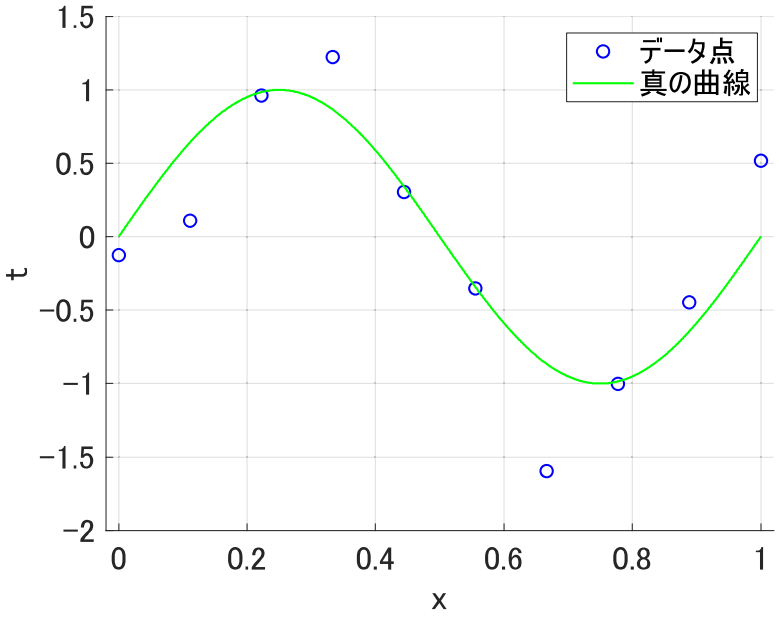


2 ：1.1節の紹介

- 1.1節 「例 : 多項式フィッティング」

1.1節で考える問題設定

- 訓練データ集合：N個の観測データ
  - 入力データ集合： $\boldsymbol{x} = (x_1, \ldots, x_N)^{\top}$
  - 目標データ集合： $\boldsymbol{t} = (t_1, \ldots, t_N)^{\top}$
- ゴール：訓練データ集合から、入出力の関係を推定して、新しい入力 $\hat{x}$ から目標変数 $\hat{t}$ を予測する。



- 本資料では、PRMLと同様の方法でデータを自ら生成した。
  以下、そのデータを用いた結果を示す。
- 入力データ：区間 [0, 1] から等間隔で N = 10 個の x_n を選ぶ。
- 出力データ：

$$\sin(2\pi x_n) + \varepsilon_n \ , \varepsilon_n \sim N(0, 0.3^2)$$

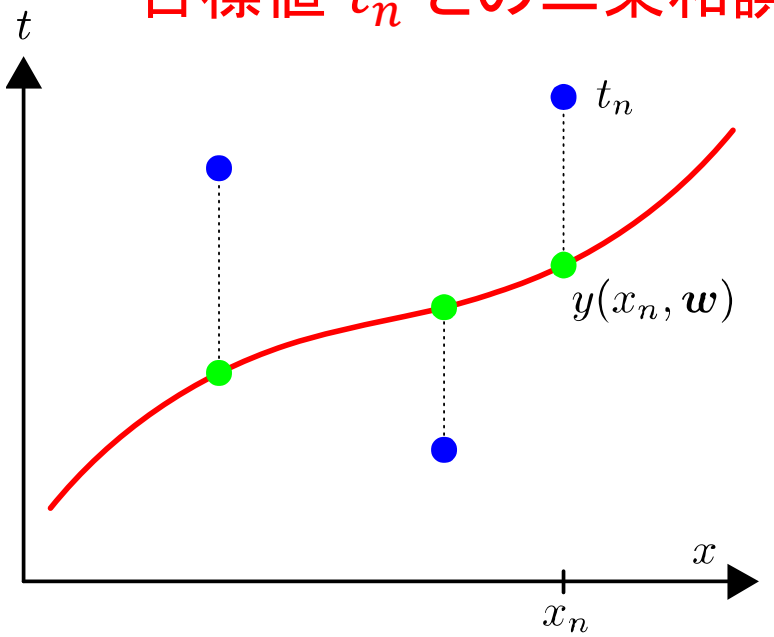
ガウス分布に従う
ランダムノイズ

多項式曲線フィッティング

- 多項式を使って、データへのフィッティングを行う。

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

- 係数 $w_0, w_1, \ldots, w_M$ を決定する方法
  →各データ点 x_n における予測値 $y(x_n, \boldsymbol{w})$ と
  目標値 t_n との二乗和誤差の最小化



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

二乗誤差の最小化問題の解の計算方法の概略

- 行列 X とベクトル w, t を次のように定める。

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^M \\ 1 & x_2 & x_2^2 & \cdots & x_2^M \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^M \end{pmatrix} \quad \boldsymbol{w} = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_M \end{pmatrix} \quad \boldsymbol{t} = \begin{pmatrix} t_0 \\ t_1 \\ t_2 \\ \vdots \\ t_M \end{pmatrix}$$

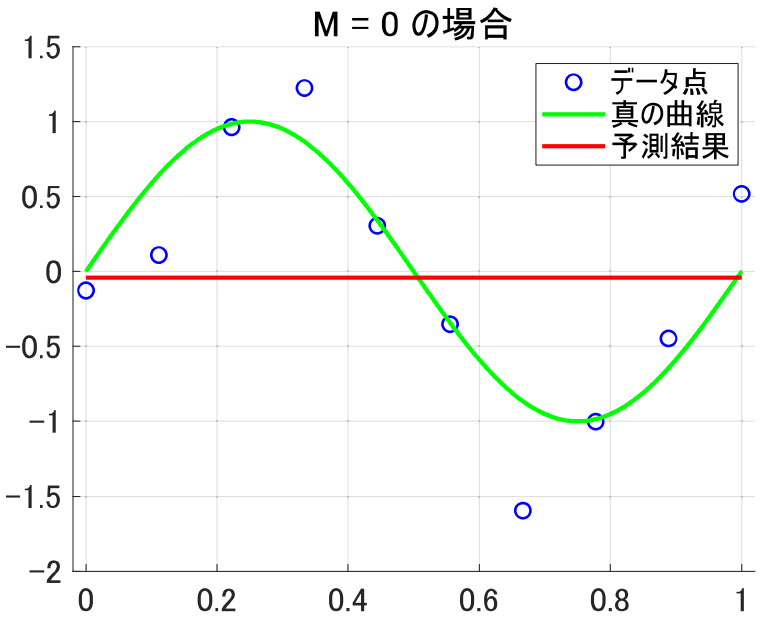
- 二乗誤差：

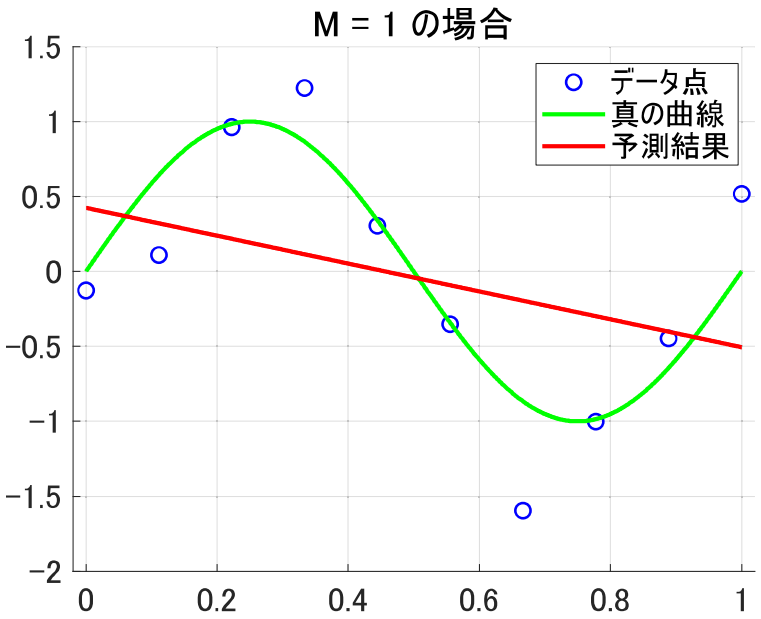
$$E(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t}\|^2$$

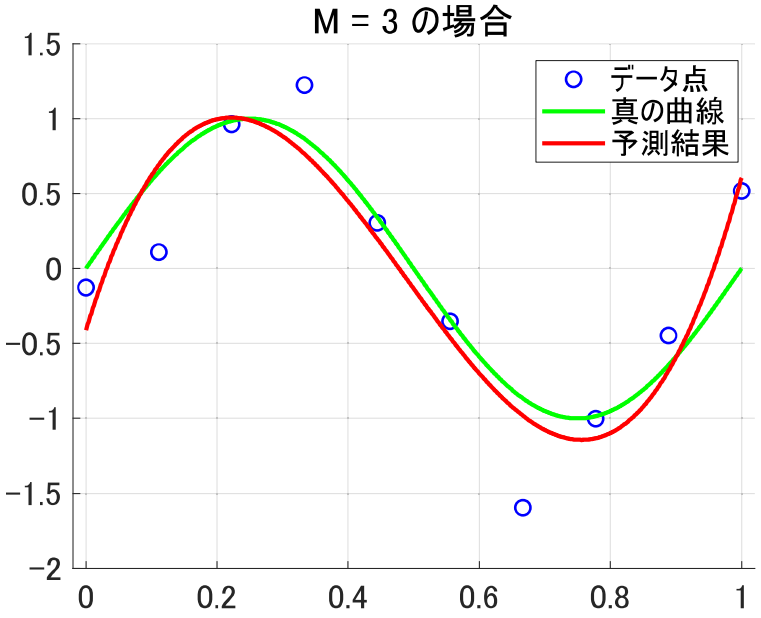
- w で微分して0となる点を求める。

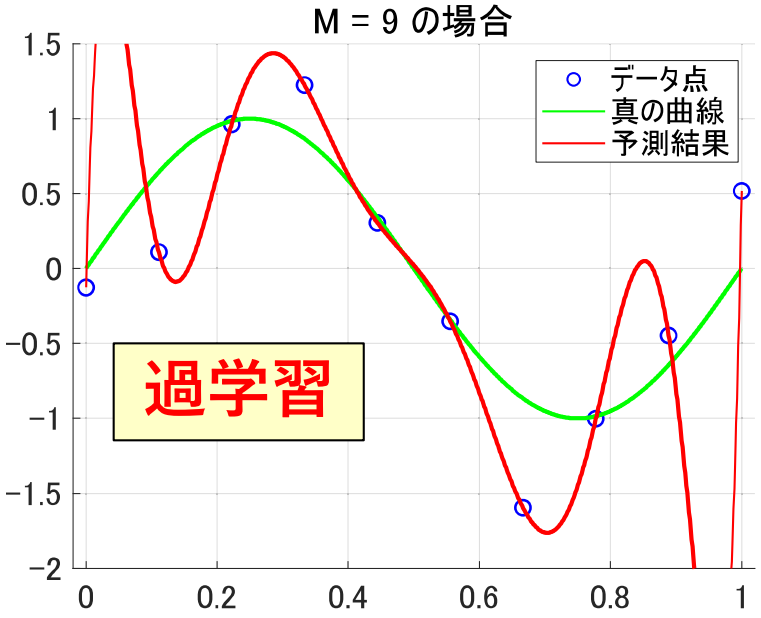
$$\frac{\partial E(\boldsymbol{w})}{\partial \boldsymbol{w}} = \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{w} - \boldsymbol{X}^\top \boldsymbol{t} \quad \Rightarrow \quad \boldsymbol{w}^* = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{t}$$

多項式曲線フィッティングの結果









過学習

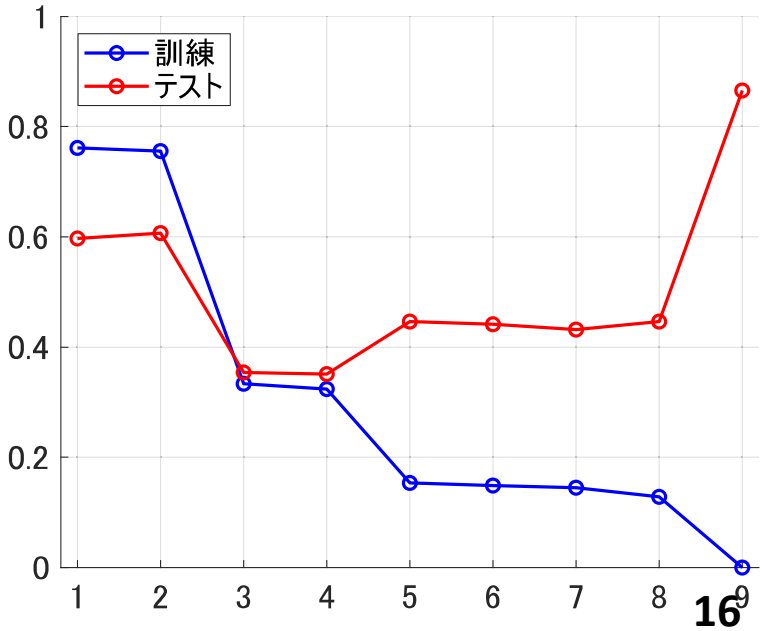
- 過学習：訓練データに非常によく当てはまっているものの、新たなデータに対してはうまく予測できない状況
  - 今回の場合、M = 0, 1 の場合は明らかに予測が不適当。
    定数や1次関数で、真の曲線 $y = \sin(2\pi x)$ を近似するのは難しい。
  - M = 3 の場合は一番よく近似できているように見える。
  - M = 9 の場合は訓練データには非常によく当てはまっているが、真の曲線 $y = \sin(2\pi x)$ の近似としては不適当。
  - M = 9 の場合、10個の重みに対して10個のデータを使用する。
    10個のデータに当てはまる9次関数を無理やり見つける。

汎化能力の評価

- 汎化能力の評価
  - 今回は、100個のテストデータを、訓練データと同様の方法で生成。
  - 汎化能力の評価指標
    → RMS error（Root Mean Square error, 平均二乗平方根誤差）

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

- テストデータに対して、
  学習で取得した重みを利用して、
  二乗誤差を計算。
- テストデータの数 N で割ることで、
  異なるサイズのデータ集合の
  比較も可能。

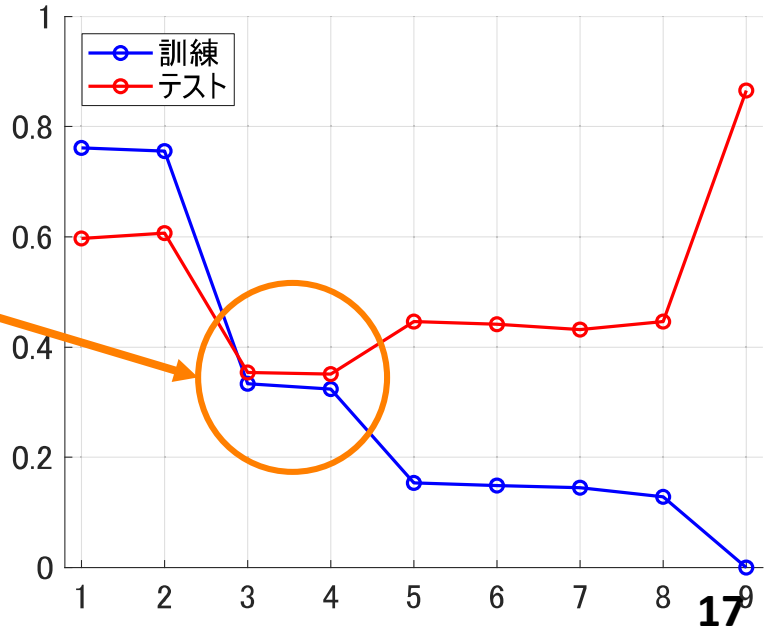


汎化能力の評価

- 汎化能力の評価
  - 今回は、100個のテストデータを、訓練データと同様の方法で生成。
  - 汎化能力の評価指標
    → RMS error（Root Mean Square error, 平均二乗平方根誤差）

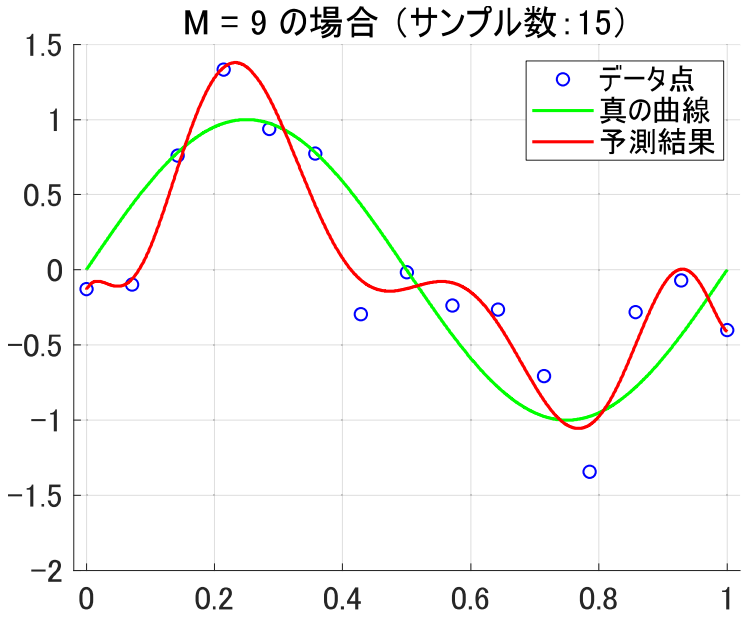
$$E_{\mathrm{RMS}} = \sqrt{\frac{2E(\boldsymbol{w}^*)}{N}} \longleftarrow E(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\Big(y(x_n, \boldsymbol{w}) - t_n\Big)^2$$

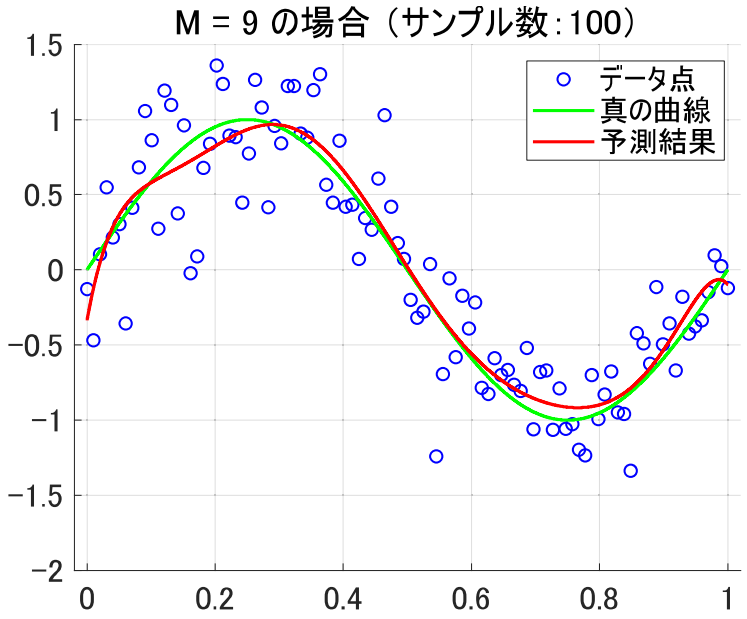
テストデータに対する当てはまりは、
M = 3, 4 の場合が一番良い。



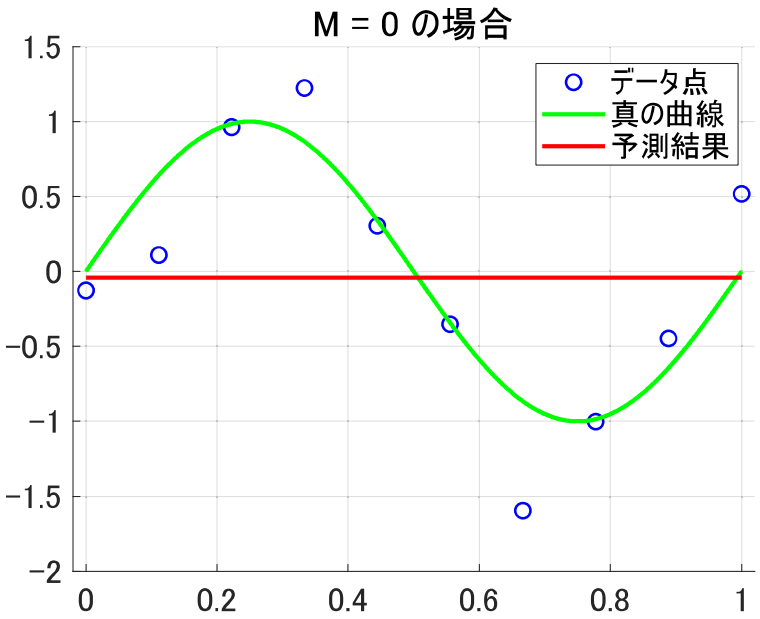
訓練データ数を増やす

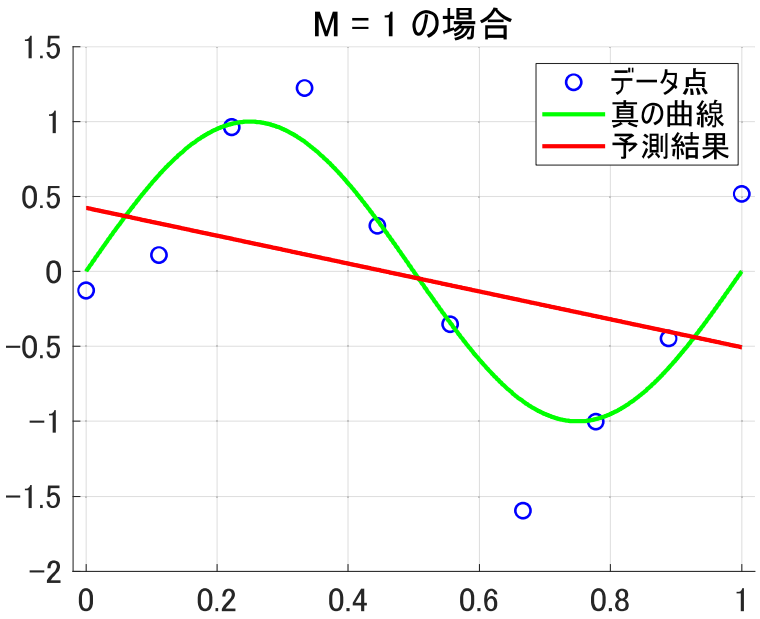
- 訓練データ数を増やせば、複雑で柔軟なモデルをデータに当てはめることができる。
- 過学習が起きないようにするには、モデル内のパラメータの数の5倍～10倍の訓練データが最低限必要となる。

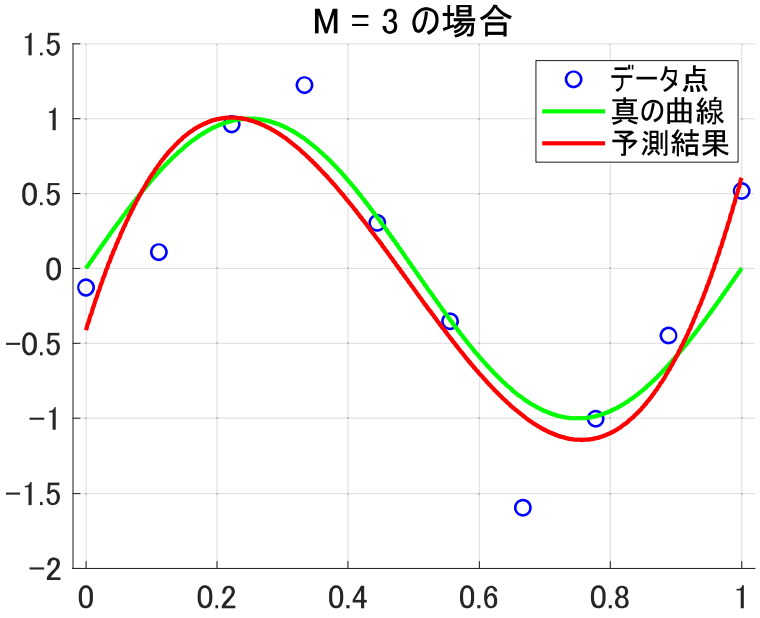


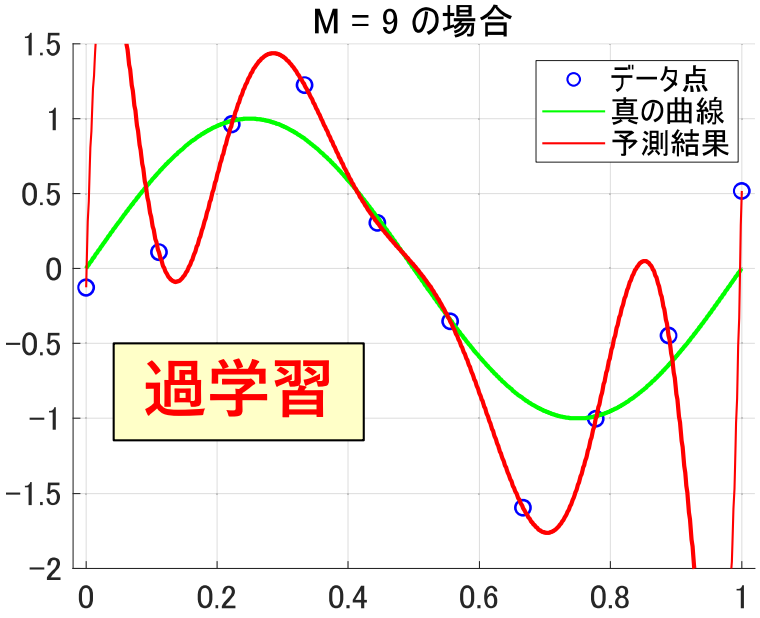


多項式曲線フィッティングの結果









重み係数の値に注目する

	M = 0	M = 1	M = 3	M = 9
0 次	-0.04	0.42	-0.41	-0.13
1 次		-0.93	14.18	158.10
2 次			-41.33	-3.76e+3
3 次			28.17	3.53e+4
4 次				-1.71e+5
5 次				4.76e+5
6 次				-7.94e+5
7 次				7.84e+5
8 次				-4.21e+5
9 次				9.50e+4

- M = 9 の場合、非常に大きな正負の値を利用して、全ての訓練データに無理やり合うように調整する。
- Mが大きく、自由度が大きくなるほど、ランダムノイズに引きずられてしまう。

正則化：リッジ回帰

- 過学習の抑制のために、
  重みが大きくなることを抑制する項を付け加える。

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

訓練データに対する適合の良さ

重みパラメータが大きくなることに対する罰則項

- λ：正則化パラメータ
  （正則化項と二乗誤差の和の相対的な重要度を調節）
- 正則化項として、重みパラメータの二乗和を用いたものは
  リッジ回帰（Ridge Regression）や l_2−正則化と呼ばれる。

リッジ回帰での重み係数の決定方法の概略

- 行列 X とベクトル w, t を次のように定める。

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^M \\ 1 & x_2 & x_2^2 & \cdots & x_2^M \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^M \end{pmatrix} \quad \boldsymbol{w} = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_M \end{pmatrix} \quad \boldsymbol{t} = \begin{pmatrix} t_0 \\ t_1 \\ t_2 \\ \vdots \\ t_M \end{pmatrix}$$

- 二乗誤差 :

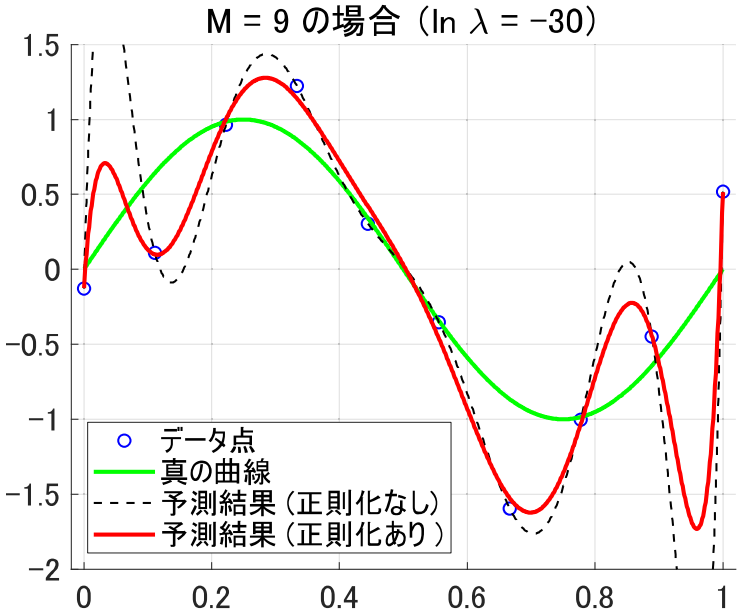
$$\widetilde{E}(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t}\|^2 + \frac{\lambda}{2}\|\boldsymbol{w}\|^2$$

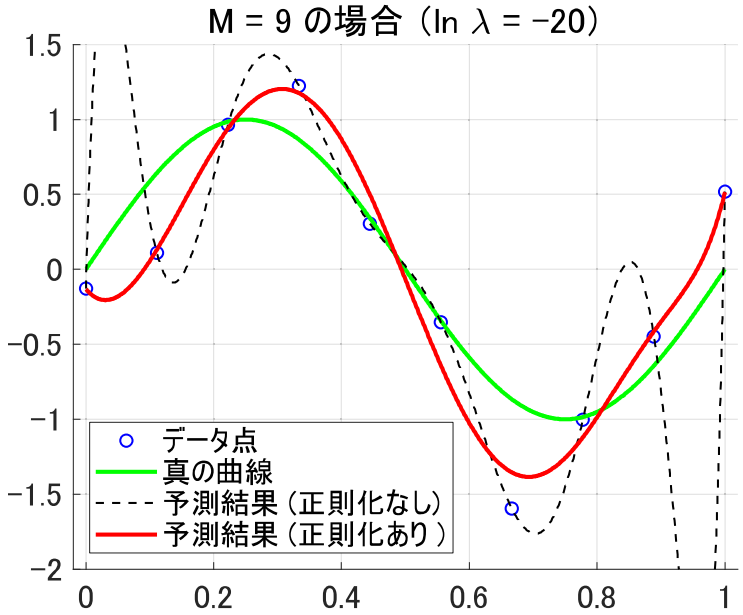
$\boldsymbol{w}$で微分して
0となる点を求める

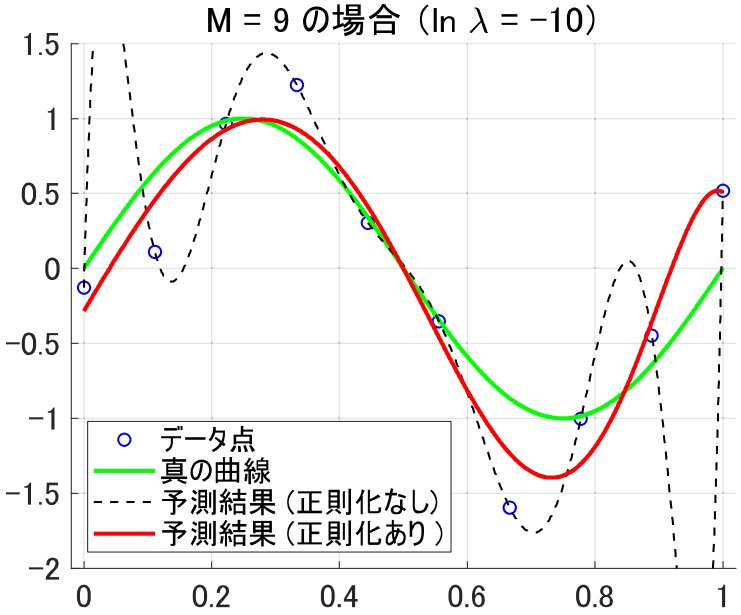
$$\frac{\partial \widetilde{E}(\boldsymbol{w})}{\partial \boldsymbol{w}} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})\boldsymbol{w} - \boldsymbol{X}^\top \boldsymbol{t}$$

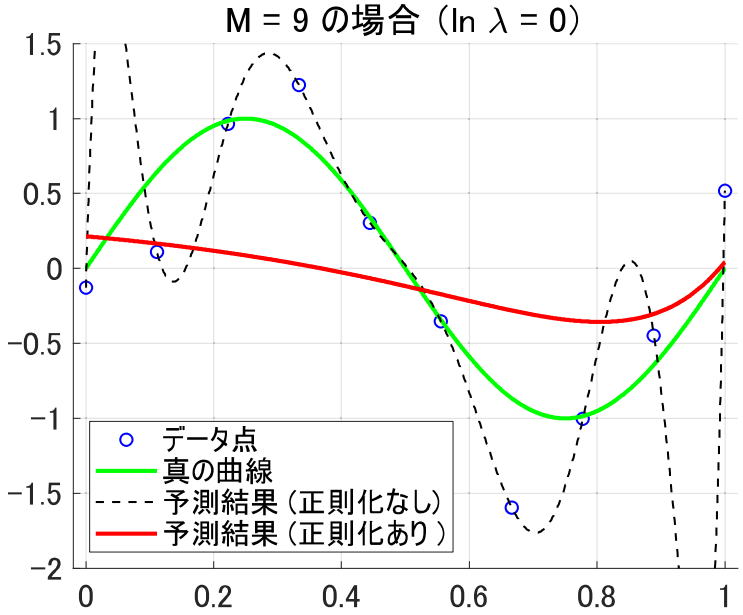
$$\boldsymbol{w}^* = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{t}$$

リッジ回帰の結果


M = 9 の場合（ln λ = −30）
データ点
真の曲線
予測結果（正則化なし）
予測結果（正則化あり）


M = 9 の場合（ln λ = −20）
データ点
真の曲線
予測結果（正則化なし）
予測結果（正則化あり）


M = 9 の場合（ln λ = −10）
データ点
真の曲線
予測結果（正則化なし）
予測結果（正則化あり）


M = 9 の場合（ln λ = 0）
データ点
真の曲線
予測結果（正則化なし）
予測結果（正則化あり）

重み係数の値（リッジ回帰の場合）

	正則化なし	$\ln \lambda = -30$	$\ln \lambda = -20$	$\ln \lambda = -10$	$\ln \lambda = 0$
0次	-0.13	-0.13	-0.13	-0.28	0.21
1次	158.10	62.32	-5.04	7.18	-0.39
2次	-3.76e+3	-1.56e+3	94.14	-1.27	-0.42
3次	3.53e+4	1.56e+4	-275.62	-29.92	-0.27
4次	-1.71e+5	-7.81e+4	278.93	-3.01	-0.10
5次	4.76e+5	2.24e+5	-299.82	18.36	0.04
6次	-7.94e+5	-3.83e+5	370.31	21.26	0.14
7次	7.84e+5	3.86e+5	267.74	11.25	0.22
8次	-4.21e+5	-2.11e+5	-818.71	-4.01	0.28
9次	9.50e+4	4.85e+4	388.72	-19.03	0.32

正則化パラメータの影響

- 正則化により、重みを抑制できている。
  → 過学習が抑制され、真の曲線をより正しく近似できる。
- 正則化パラメータが大きすぎると、重みを抑制しすぎて、
  訓練データにうまく当てはまっていない。
- 正則化パラメータの選び方は、非常に重要。
  モデルへの当てはまりの程度や汎化能力に影響を与える。