機械学習①

回帰

東京大学　大学院工学系研究科

吉元俊輔

やりたいこと

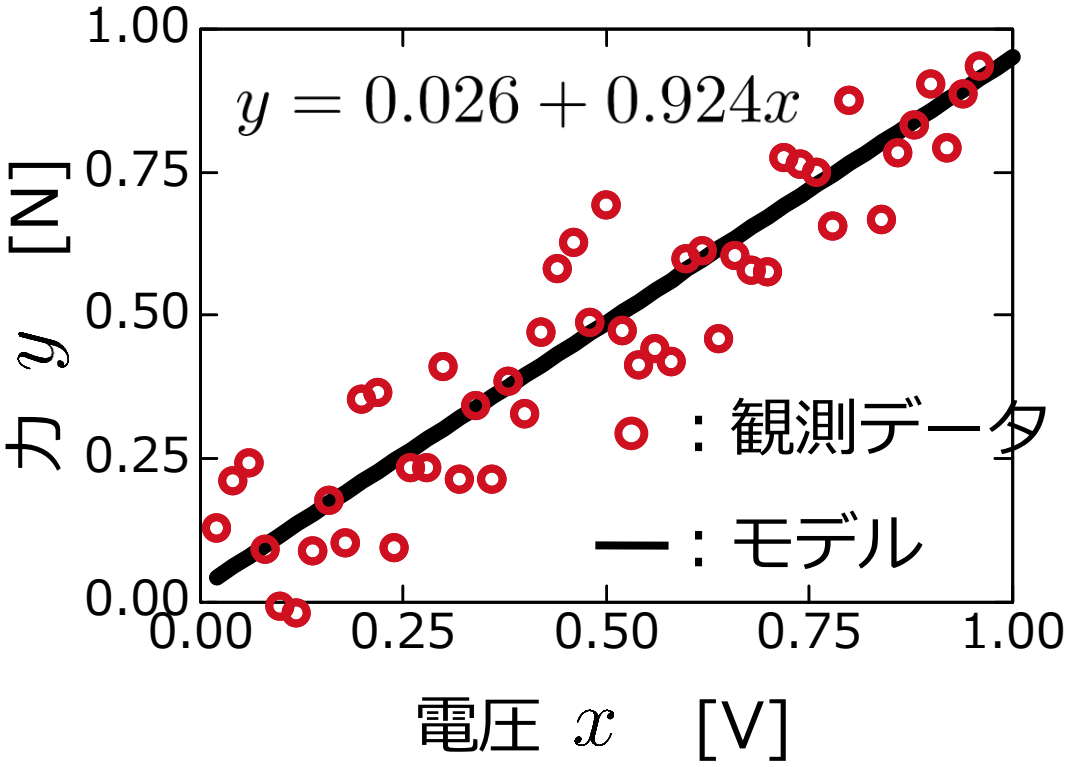
- 連続変数の関係をモデル化し，一方から他方を求める
- 例えば,
  - 計測された電圧から力の値を求める
  - 複数の無線ルータの電波を受信した強度から位置を推定する
  - ステレオカメラから得られた視差画像から3次元形状を推定する
- など，単数ないし複数の入出力とそのモデルを扱う
- ただし,
  - 過去10年の株価変動から未来の株価を推定する
  - 細胞分裂の進行の様子から未来の細胞数を推定する
- など，時系列データに関しては範囲外とする

用語の説明

- 説明（独立）変数・目的（従属）変数
  - 変数xが，変数tを用いて数式表現できる時，tを説明変数，xを目的変数という
- モデル選択（モデル化，モデル比較）
  - 説明変数と目的変数の関係を数式表現することをモデル選択という
- 入力（観測）変数・出力（目標）変数
  - モデルにおいて，入力（観測）される変数を入力変数，所望の値を出力変数という
- 学習（訓練）
  - 観測されたデータに基づき，モデルパラメータを最適化することを学習という
- 教師あり・なし
  - 学習において，入出力が与えられるのを教師あり，入力データのみが与えられるのを教師なしという
- 回帰・クラス分類
  - 入力変数から，連続的な出力変数を求める問題を回帰という
  - 入力変数から，有限個のカテゴリに分ける問題をクラス分類という

具体的に見てみる（電圧から力を予測する）

- 連続変数を扱う問題なので回帰
- 次数1の線形回帰モデルを最小二乗法でパラメータ最適化



x ：入力変数（電圧）

y ：出力変数（力）

$y = w_0 + w_1 x$

：線形回帰モデル

$\mathbf{w} = (w_0, w_1)^{\mathrm{T}}$

：モデルパラメータ

回帰の手順

- 学習（訓練）
  - 入出力変数を明らかにする
  - モデル選択（比較）を行う
  - 学習用の入出力データを取得する
  - モデルパラメータの最適化を行う
- 予測（推定）
  - 得られたパラメータを用いて入力から出力を計算する
- 評価（テスト）
  - テスト用のデータを取得し，評価指標を計算する

入出力変数を明らかにする

- 出力変数
  - 目的に応じて必然的に決まる
  - 複数存在する場合もある：次元数
- 入力変数
  - 観測可能な情報から得られなければならない
  - 観測信号がそのまま入力変数になるとは限らない：特徴量抽出
  - 出力と入力変数の間に因果関係が必要である
  - 複数存在する場合もある：次元数
  - 過剰に用意すると無駄である：基底ベクトル
  - 不足しているとうまく推定できない：不確定性

モデル選択を行う

- 入出力変数の関係を表す：単純に，多項式の線形和で表現する

x: 入力変数　　y ：出力変数　　$\mathbf{w} = (w_0, ..., w_M)^{\mathrm{T}}$ ：パラメータ

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

- パラメータは誤差関数の最小化で求めることができる：最適化
- 誤差関数の代表例：予測値　$y(x_n, \mathbf{w})$ と真値　t_n の二乗和誤差

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

β：精度パラメータ

- 多項式の次数　M（モデルの複雑さ）を選ぶ：評価

入力変数が複数ある場合

三次の項まで考えると下記のようになる

$\mathbf{x} = (x_1, ..., x_D)^{\mathrm{T}}$: 入力変数（ D 次元）

y : 出力変数

$\mathbf{w} = (w_0, w_1..., w_D, w_{11}, ..., w_{DD}, w_{111}, ..., w_{DDD})^{\mathrm{T}}$

: パラメータ

$$y = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D}\sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D}\sum_{j=1}^{D}\sum_{k=1}^{D} w_{ijk} x_i x_j x_k$$

D 次元の入力で M 次の多項式を考えると $\sum_{m=0}^{M} D^m$ 項になる

入力と、出力変数も複数ある場合

三次の項まで考えると下記のようになる

$\mathbf{x} = (x_1, ..., x_D)^{\mathrm{T}}$ ：入力変数（ D 次元）

$\mathbf{y} = (y_1, ..., y_K)^{\mathrm{T}}$ ：出力変数（ K 次元）

$\mathbf{W} = (\mathbf{w}_0, \mathbf{w}_1 ..., \mathbf{w}_D, \mathbf{w}_{11}, ..., \mathbf{w}_{DD}, \mathbf{w}_{111}, ..., \mathbf{w}_{DDD})$

：パラメータ（ $\sum_{m=0}^{M} D^m \times K$ 回帰係数行列）

$$\mathbf{y} = \mathbf{w}_0 + \sum_{i=1}^{D} \mathbf{w}_i x_i + \sum_{i=1}^{D}\sum_{j=1}^{D} \mathbf{w}_{ij} x_i x_j + \sum_{i=1}^{D}\sum_{j=1}^{D}\sum_{k=1}^{D} \mathbf{w}_{ijk} x_i x_j x_k$$

の部分を一般化 $\phi(\mathbf{x})$ して $\mathbf{y} = \mathbf{W}^{\mathrm{T}}\phi(\mathbf{x})$のように表すと単純

入力変数に対して非線形に振る舞う場合は・・・

- 入力変数の多項式で表すのをやめて一般化する
- 基底関数 $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), ..., \phi_M(\mathbf{x}))^{\mathrm{T}}$ の線形和で表す

（二次多項式の場合は $\phi(\mathbf{x}) = (1, x_1, ..., x_D, x_1x_1, ..., x_Dx_D)^{\mathrm{T}}$ ）

- 線形回帰モデルは　$\mathbf{y} = \mathbf{W}^{\mathrm{T}}\phi(\mathbf{x})$
- 基底関数＝変数変換による非線形化が可能

$\phi_j(x) = x^j$ ：スプライン関数

$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$ ：ガウス関数

$\phi_j(x) = \frac{1}{1+\exp\left\{-\frac{x-\mu_j}{s}\right\}}$ ：ロジスティックシグモイド関数

（フーリエ基底・ウェーブレットも重要）

学習用の教師データを取得する

- データの計測範囲を決める
  - 上限下限を問題設定に応じて決める
  - 入出力の変数値が変化する範囲で取得する
- データのサンプル数を決める
  - データは多いほどモデルの信頼性は向上するが計算量が増える
  - 経験的にはパラメータの数の5～10倍のサンプル点があればよい
  - データが少なすぎると，パラメータが一意に決まらない
  - 計測範囲をまんべんなくサンプルする
- 入出力変数以外のパラメータを出来る限り固定して（不確定要素を減らして）データを取得するべし

モデルパラメータの最適化を行う

- 最尤推定法
  - 尤もらしい条件を見つける方法：尤度関数の最大化
  - 複雑なモデルであればあるほど尤度関数を大きくする
  - 最小二乗法：二乗誤差和を最小化するパラメータを見つける
  - 正則化最小二乗法：罰則（正則化）項を誤差関数に加える
- ベイズ推定法（詳しい説明は省略）
  - 最尤推定法に加え，事前確率を導入し，パラメータを見つける
  - コントロール出来ないパラメータを事前確率として考慮できる
  - マルコフ連鎖モンテカルロ法を使って数値計算する事が可能

モデルで推定できる値は真値なのか？

- 推定される値には不確実性がある
  - 計測データに規則性のないノイズが含まれる

    観測条件を統制しきれないか，確率的現象を含む問題を扱っている
  - 学習データが有限であるためモデルが不完全である
- 誤差関数（多次元入力1次元出力，一般化表現の場合）
  - 予測値　$y = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)$　と真値　t_n の二乗和誤差

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left\{\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) - t_n\right\}^2$$

データの分布と不確実性

- 正規分布（ガウス分布）

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

x ：変数　μ ：平均　σ ：分散

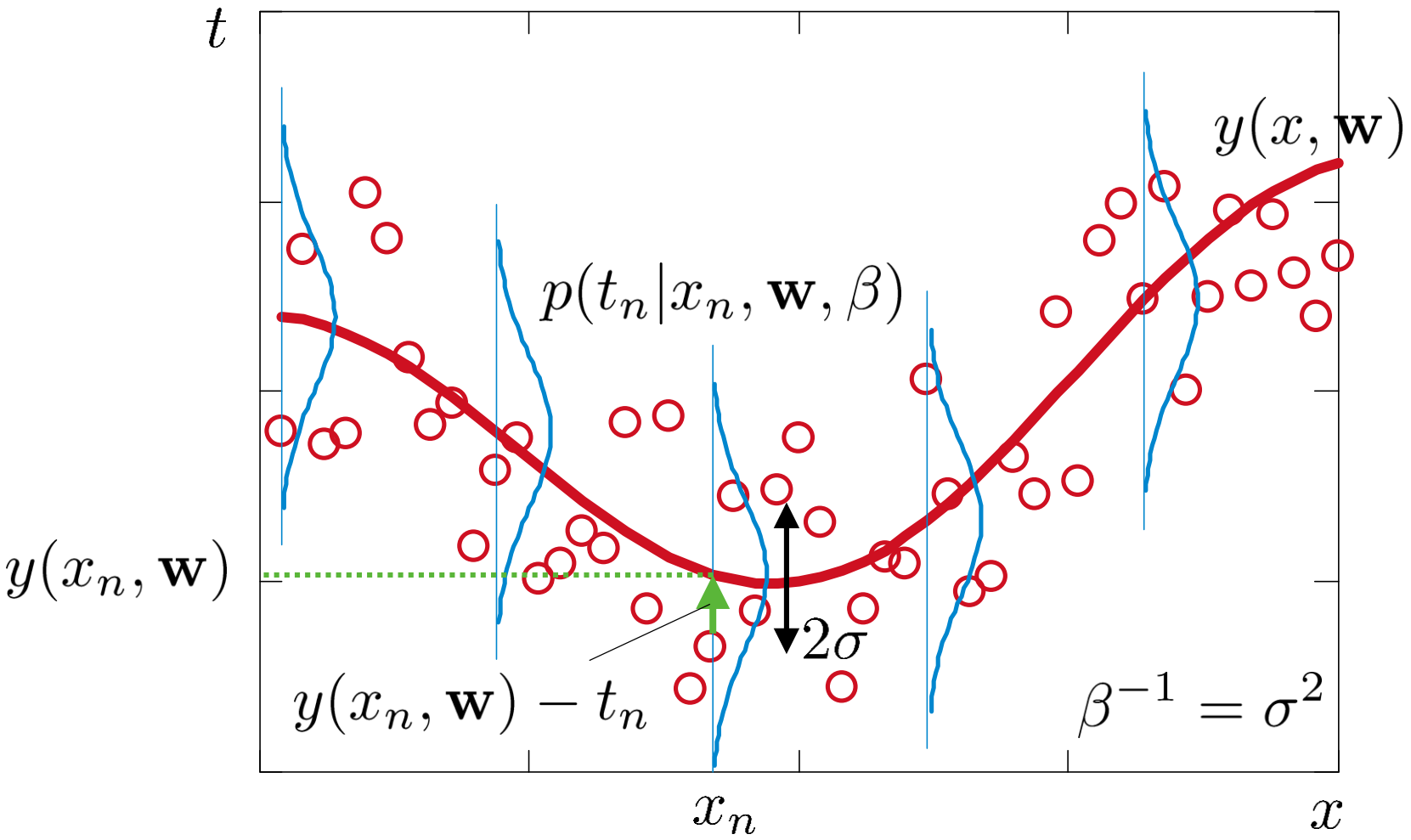
- 不確実性を正規分布で表現する：尤度関数

$$p(\mathbf{t}|\mathbf{X},\mathbf{w},\beta) = \prod_{n=1}^{N}\mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n),\beta^{-1})$$

$\mathbf{X} = \{\mathbf{x}_1,...,\mathbf{x}_N\}$ ：観測値　　$\mathbf{t} = \{t_1,...,t_N\}$ ：目標

$p(Y|X)$ ：X が与えられた下でのY の確率

つまり、観測値に対して真値が確率的に決まる



最小二乗法

- 対数尤度関数を最大にするようなパラメータを見つける

  ＝二乗和誤差関数の最小化問題

$$\ln p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \frac{N}{2}\ln \beta - \frac{N}{2}\ln(2\pi) - \frac{\beta}{2}\sum_{n=1}^{N}\left\{\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) - t_n\right\}^2$$

- パラメータでの偏微分を求める

$$\nabla \ln p(\mathbf{t}|\mathbf{w}, \beta) = \beta \sum_{n=1}^{N}\left\{\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) - t_n\right\}\phi(\mathbf{x}_n)^{\mathrm{T}}$$

- 偏微分した式がゼロとなるようなパラメータを計算する

  ＝連立方程式を解く

$$\mathbf{w}_{\mathrm{ML}} = (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t} \qquad \mathbf{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

擬似逆行列

1次元入出力、1次関数モデルの最小二乗法

x ：入力　　t ：観測データ　$y = w_0 + w_1 x$ ：モデル式

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{w_0 + w_1 x_n - t_n\}^2$$ ：誤差関数

- 偏微分してゼロと置いた式　→　連立方程式

$$\frac{\partial E(\mathbf{w})}{\partial w_0} = \sum_{n=1}^{N}\{w_0 + w_1 x_n - t_n\} = 0$$

$$\frac{\partial E(\mathbf{w})}{\partial w_1} = \sum_{n=1}^{N}\{w_0 + w_1 x_n - t_n\} x_n = 0$$

$$\begin{pmatrix} N & \sum x_n \\ \sum x_n & \sum x_n^2 \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \end{pmatrix} = \begin{pmatrix} \sum t_n \\ \sum x_n t_n \end{pmatrix}$$

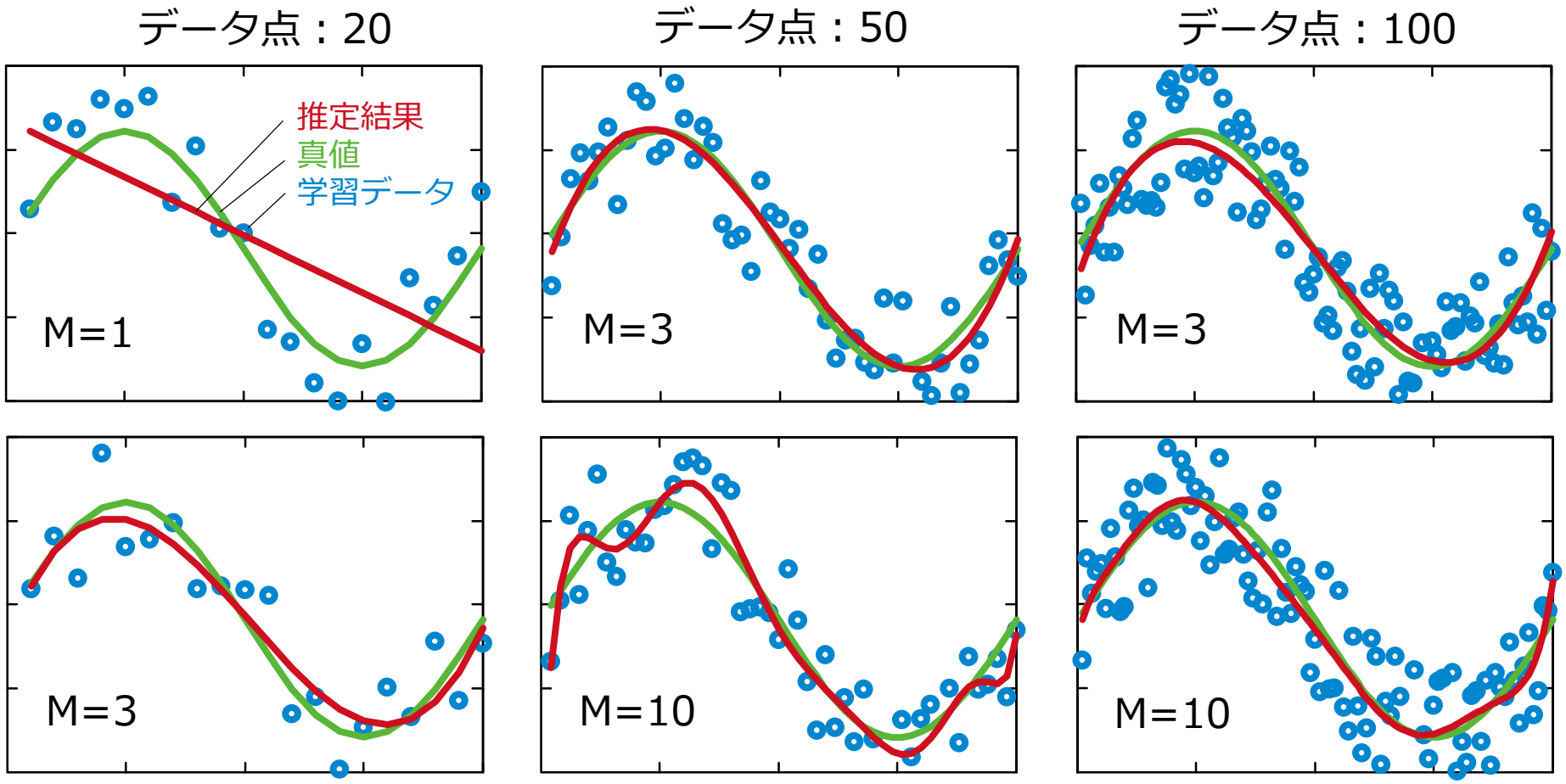
$$\begin{pmatrix} w_0 \\ w_1 \end{pmatrix} = \begin{pmatrix} N & \sum x_n \\ \sum x_n & \sum x_n^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum t_n \\ \sum x_n t_n \end{pmatrix}$$

$$= \frac{1}{N\sum x_n^2 - (\sum x_n)^2} \begin{pmatrix} N\sum x_n t_n - \sum x_n \sum t_n \\ N\sum x_n^2 \sum t_n - \sum x_n \sum x_n t_n \end{pmatrix}$$

単純な例で最小二乗法の結果を見てみる

学習データ：正弦波にランダムノイズを重畳

$$\phi_j(x) = x^j \qquad j = 0, \ldots, M$$



Rで確認するためのコード

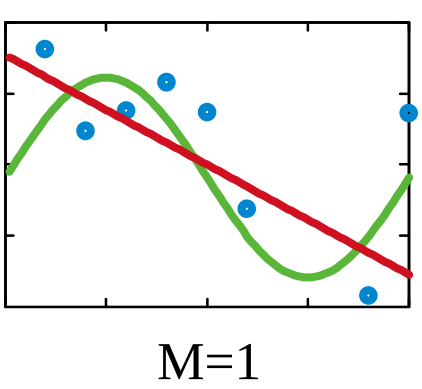
```
L<-100
x<-(c(1:L))/(L)
y<-(sin(2*pi*x)+1.3)*0.35
y_rand<-y+runif(L, min=-0.2, max=0.2)
Dim<-3
phi<-matrix(nrow=Dim+1,ncol=L)
for(i in 1:(Dim+1)){
  phi[i,]<-x^(i-1)
}
w<-y_rand%*%t(phi)%*%solve(phi%*%t(phi))
y_est<-w%*%phi

plot(x,y_rand,type="p" ,col=rgb(0,0,1))
par(new=T)
plot(x,y,type="l",col=rgb(0,1,0))
par(new=T)
plot(x,y_est,type="l",col=rgb(1,0,0))
```

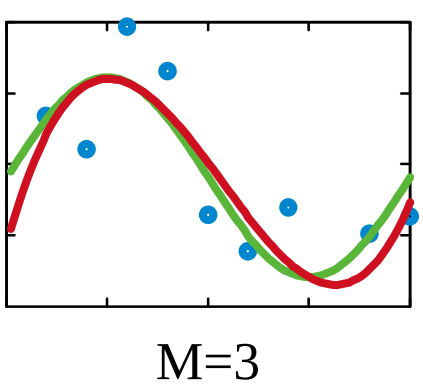
評価：どのくらいうまく推定できるか

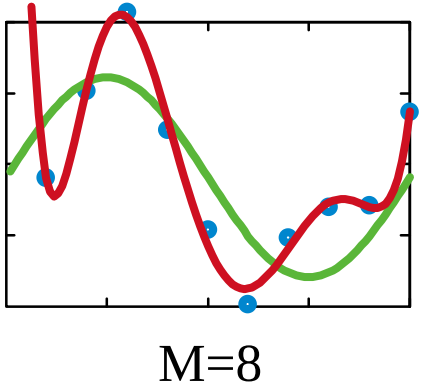
- 学習データで取得したモデルを評価する
  - 測定データの一部を学習，一部をテイストに使用する：交差確認
  - 評価指標を計算する：代表例が平均二乗平方根誤差

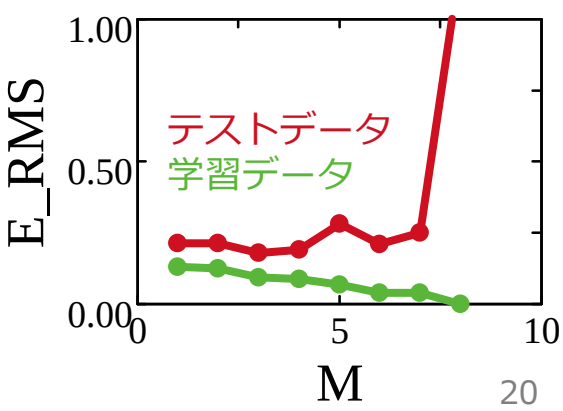
$$E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w})/N} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\{\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) - t_n\}^2}$$

- 学習データにはよく合うけどテストデータはだめ：過学習









正則化で過学習を防ぐ

- モデルの複雑さに罰則を科す
  - 滑らかでないことに罰則をかける
  - パラメータのノルムの大きさに罰則をかける
- 代表例：ノルムの大きさを罰則として誤差関数に加える

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left\{\mathbf{w}^{\mathrm{T}}\phi(x_n) - t_n\right\}^2 + \frac{\lambda}{2}||\mathbf{w}||^2$$

$$||\mathbf{w}||^2 \equiv \mathbf{w}^{\mathrm{T}}\mathbf{w} = w_0^2 + \cdots + w_{M-1}^2$$

- この場合の最小二乗解は下記で表される

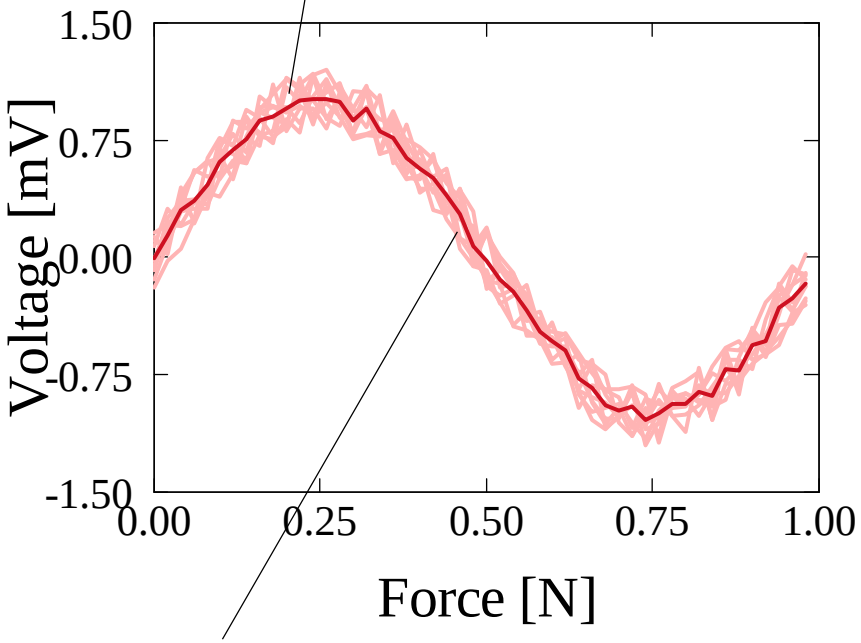
$$\mathbf{w}_{\mathrm{ML}} = (\lambda\mathbf{I} + \mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}$$

外れ値の影響

- 最小二乗法は，モデルと観測値の誤差が平均0の正規分布に従う場合は最適なモデルを推定する
- 外れ値の影響を受けないようにするには：ロバスト推定
  - 目視で外れ値を除外
  - 誤差の評価方法を変更
    - M-estimator
    - LMedS(最小二乗メディアン)推定
  - 一部の(よくモデルに適合する)データから推定
    - RANSAC (RANdom SAmple Consensus)

グラフの載せ方

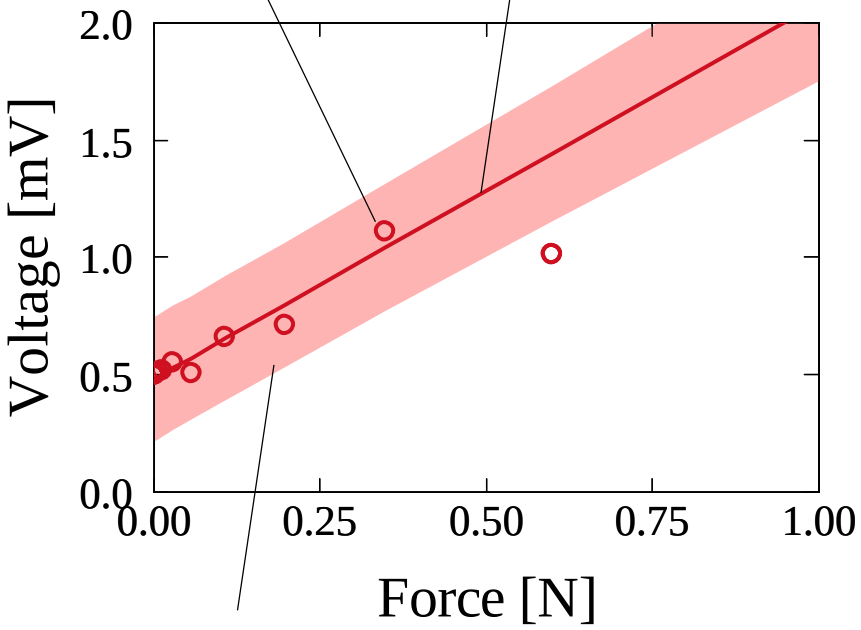
繰り返し計測したデータを薄く線でプロット



平均値を濃く線でプロット

データを点でプロット

モデルの曲線をプロット



予測区間を薄い色で網掛け

$$\frac{1}{\beta_{\mathrm{ML}}}$$

予測区間：推定した値は95%（一般的には）の確率でこの範囲に入る

自分の問題がどれに当たるかを把握しておく

- 変数の数は？
- どれが入力でどれが出力か？
- モデルはどうやって選ぼうか？
  - 理論的に定式化できる場合はそれを優先的に利用する
  - 理論を構築することが困難な場合は基底関数の線形和を考える
- 記号が何か明記する。
- ベクトルや行列の次元を明記する。

多次元入力，1次元出力の場合

入力変数（D次元）　：$\mathbf{x}_n = (x_{1,n} \ \cdots \ x_{D,n})^{\mathrm{T}}$

基底関数（M次元）　：$\boldsymbol{\phi}(\mathbf{x}_n) = (\phi_1(\mathbf{x}_n) \ \cdots \ \phi_M(\mathbf{x}_n))^{\mathrm{T}}$

観測変数（１次元）　：y_n

推定変数（１次元）　：$\hat{y}_n$

モデル式　：$\hat{y}_n = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)$

観測データ（N次元）　：$\mathbf{y} = (y_1 \ \cdots \ y_N)$

基底行列（$M \times N$次元）　：$\boldsymbol{\Phi} = \{\boldsymbol{\phi}(\mathbf{x}_1) \ \cdots \ \boldsymbol{\phi}(\mathbf{x}_N)\}$

最尤パラメータ（M次元）　：$\mathbf{w}_{\mathrm{ML}} = (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{y}^{\mathrm{T}}$

$\mathbf{w}_{\mathrm{ML}}^{\mathrm{T}} = \mathbf{y}\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}})^{-1}$

多次元入力，多次元出力の場合

入力変数（D次元）　：$\mathbf{x}_n = (x_{1,n} \ \cdots \ x_{D,n})^{\mathrm{T}}$

基底関数（M次元）　：$\boldsymbol{\phi}(\mathbf{x}_n) = (\phi_1(\mathbf{x}_n) \ \cdots \ \phi_M(\mathbf{x}_n))^{\mathrm{T}}$

観測変数（L次元）　：$\mathbf{y}_n = (y_{1,n} \ \cdots \ y_{L,n})^{\mathrm{T}}$

推定変数（L次元）　：$\hat{\mathbf{y}}_n = (\hat{y}_{1,n} \ \cdots \ \hat{y}_{L,n})^{\mathrm{T}}$

モデル式　：$\hat{\mathbf{y}}_n = \mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)$

観測データ（$L \times N$次元）　：$\mathbf{Y} = \{\mathbf{y}_1 \ \cdots \ \mathbf{y}_N\}$

基底行列（$M \times N$次元）　：$\boldsymbol{\Phi} = \{\boldsymbol{\phi}(\mathbf{x}_1) \ \cdots \ \boldsymbol{\phi}(\mathbf{x}_N)\}$

最尤パラメータ（$M \times L$ 次元）　：$\mathbf{W}_{\mathrm{ML}} = (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}$

$\mathbf{W}_{\mathrm{ML}}^{\mathrm{T}} = \mathbf{Y}\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}})^{-1}$

パラメータ推定から評価まで

1. 学習用データを各変数に格納する
2. 最尤パラメータを計算する
3. （必要に応じて最尤パラメータを可視化する）
4. 学習データで最小二乗誤差を計算する
5. テストデータで推定を行う
6. （必要に応じて推定結果をプロットする）
7. テストデータで最小二乗誤差を計算する

参考図書

- C.M.ビショップ 著，パターン認識と機械学習（PRML），丸善出版，東京，2012.
- 市原清志著，バイオサイエンスの統計学，南江堂，東京，2013.
- D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters." Journal of the Society for Industrial & Applied Mathematics, 11(2): 431-441, 1963.