線形回帰モデル (PRML第三章)

須加　拓

線形回帰モデル

回帰

データから関数を求める

（例）：気象データ（気圧、気温、降水量など）と
顧客のデータからスーパーの売上を出力する
関数を求める

目的

未来を予測したい！！

では、どんな問題を解くか？

入力 **目的**

$(x_1, \dots, x_D)^T$ と t

訓練データ・・・N個あるとする

$\mathbf{x}_n$ t_n ← n個目の訓練データ

この時、$\mathbf{x}_n$とt_nは

$$t_n = y(\mathbf{x}_n) + \epsilon$$ ← **ノイズ**

という関係があるとする

回帰

訓練データから$y(\mathbf{x})$をもとめる

y(**x**)はどんな形？

$y(\mathbf{x})$のモデルとして最も単純なモデルは

$$y(\mathbf{x},\mathbf{w}) = \omega_0 + \omega_1 x_1 + \cdots + \omega_D x_D$$

しかし、これでは入力に関しても線形関数になっているので、表現能力が乏しい

そこで、非線形な関数の線形結合を考える

$$y(\mathbf{x},\mathbf{w}) = \omega_0 + \sum_{j=1}^{M-1} \omega_j \boxed{\phi_j(\mathbf{x})} = \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}).$$

基底関数

線形回帰

wの値を決めることと基底関数の数Mを決めることと基底関数の中身を決めること

基底関数の種類

多項式基底関数

$$\phi_j(x) = x^j$$

ガウス基底関数

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

ロジスティックシグモイド関数

$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right) \qquad \sigma(a) = \frac{1}{1+\exp(-a)}$$

しかし、本性で示すほとんどの事項は基底関数の選択に依存しないため、基底関数を具体的に限定しないこととする

では、問題をどう解くか？

①: wの値を決めること
②: 基底関数の数Mを決めること

パラメータwはひとつの値を持っている。だから、その値を見つけていこう

頻度主義

パラメータwの値はひとつに決まっているのではなく、ある程度幅を持っている。だから、パラメータwの分布を考えていこう

ベイジアン

頻度主義で解く！

パラメータwのもつ値を求めたい

どうやって？

最尤推定(尤度を最大にするw を求める)をする

尤度

観測データはすべて出尽くしていて，
それらのデータに対して，あるパラメータの
確率分布を当てはめた時，
どれだけ尤もらしいかを意味している

最尤推定をしていく！

その前に問題を思い出そう・・・

$\mathbf{x}_1$とt_1　…　$\mathbf{x}_n$とt_n　…　$\mathbf{x}_N$とt_N　訓練データ

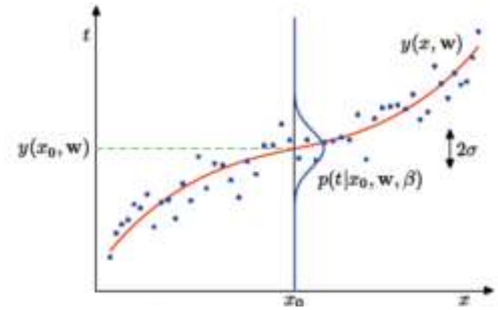
$\mathbf{x}_n$とt_nの関係は

$$t_n = y(\mathbf{w}, \mathbf{x}_n) + \epsilon$$

ノイズをどう決めるか？出来れば、予測値に近いと確率が高くて,
離れるほど確率が低くなる確率分布に従って欲しい。
そこで、ϵは期待値が0で精度がβのガウス確率変数であるとする。

$$p(t_n \mid \mathbf{x}_n, \underline{\beta}) = \mathcal{N}(t_n \mid y(\mathbf{x}_n), \beta^{-1})$$

未知



最尤推定をしていく！！

このとき、尤度関数は

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

ただし、$\mathbf{X}=\{\mathbf{x}_1,\dots,\mathbf{x}_N\}$とする

計算しやすいように対数をとると、

$$\ln p(\mathbf{t}\,|\,\mathbf{w},\beta)=\sum_{n=1}^{N}\ln N(t_n\,|\,\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}_n),\beta^{-1})$$

$$=\frac{N}{2}\ln\beta-\frac{N}{2}\ln(2\pi)-\beta\boxed{\frac{1}{2}\sum_{n=1}^{N}\{t_n-\mathbf{w}^T\phi(x_n)\}^2}$$

二乗和誤差関数になっている！

最尤推定をしてwを求める！

対数尤度関数をwで偏微分する

$$\ln p(\mathbf{t}|\mathbf{w},\beta) = \boxed{\frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi)} - \beta\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(x_n)\}^2$$

定数

ここを最小化すればいい！

$$\nabla \ln p(\mathbf{t}|\mathbf{w},\beta) = \beta\sum_{n=1}^{N}\{t - \mathbf{w}^T\phi(x_n)\}\phi(x_n)$$

=0とおいてwについてとくと,

$$\mathbf{w}_{ML} = \boxed{(\Phi^T\Phi)^{-1}\Phi^T}\mathbf{t}$$

ムーア・ペンローズの擬似逆行列

ただし $\Phi = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$

最尤推定をしてβを求める！

対数尤度関数は

$$\ln p(\mathbf{t}|\mathbf{w},\beta)=\sum_{n=1}^{N}\ln N(t_n|\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}_n),\beta^{-1})$$

$$=\frac{N}{2}\ln\beta-\frac{N}{2}\ln(2\pi)-\beta\frac{1}{2}\sum_{n=1}^{N}\{t_n-\mathbf{w}^T\phi(x_n)\}^2$$

これをβに関して偏微分したものを0とおくと、

$$\frac{1}{\beta_{ML}}=\frac{1}{N}\sum_{n=1}^{N}\{t_n-\mathbf{w}_{ML}{}^T\boldsymbol{\varphi}(\mathbf{x}_n)\}^2$$

求めたい値が手に入った！！バンザイ！！終わり！！

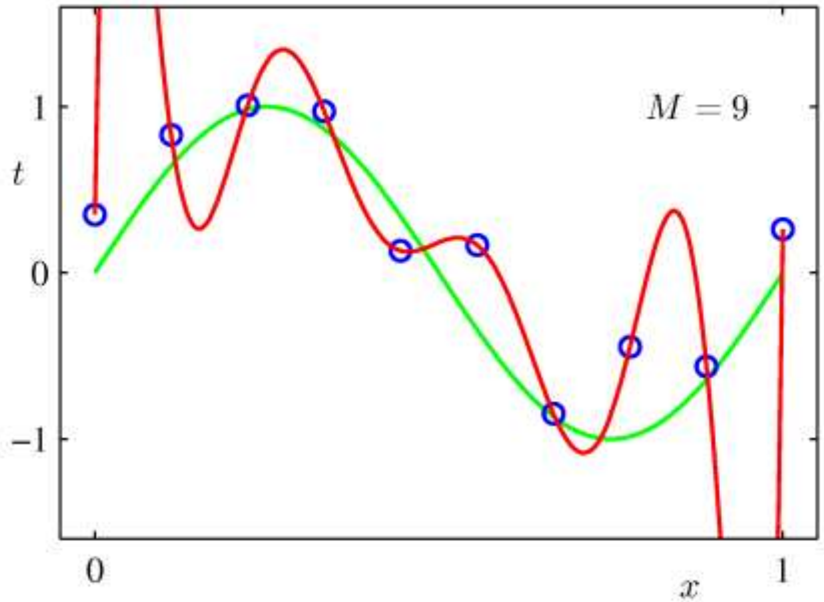
と、簡単に終わるわけもなく、最尤推定には過学習という弱点がついてくる

過学習とは？

$$y(\mathbf{x}, \mathbf{w}) = \omega_0 + \cdots + \omega_j \phi_j(\mathbf{x}) + \cdots$$

基底関数の数を増していけば、表現能力（自由度）は増していく
しかし、訓練データの数に対してあまりに多くの基底関数を使うと、その増した表現力で無理にデータに合わせようとしてしまい、実際求めている関数とはかけ離れた答えになってしまう

この過学習が起きているとき、係数の絶対値が概ねとても大きくなっています。そこで、係数が大きくなりすぎたらペナルティがつくように変えてみることを考える



$$y = 0.35 + 232.37x - 5321.83x^2 + 48568.31x^3 - 231639.30x^4 + 640042.26x^5 - 1061800.18x^6 + 1042400.18x^7 - 557682.99x^8 + 125201.43x^9$$

ペナルティ項を考えるその前に･･･①

最小二乗法の幾何学的解釈を考えよう

$y(\mathbf{x},\mathbf{w}) = \mathbf{w}^T\phi(\mathbf{x}_n)$なので

$$\mathbf{y} = \begin{pmatrix} y(\mathbf{x}_1,\mathbf{w}) \\ y(\mathbf{x}_2,\mathbf{w}) \\ \vdots \\ y(\mathbf{x}_N,\mathbf{w}) \end{pmatrix}$$ は $$\varphi_j = \begin{pmatrix} \phi_j(\mathbf{x}_1) \\ \phi_j(\mathbf{x}_2) \\ \vdots \\ \phi_j(\mathbf{x}_N) \end{pmatrix}$$ $j \in \{0,\ldots M-1\}$で張られる線形部分空間S上にある.

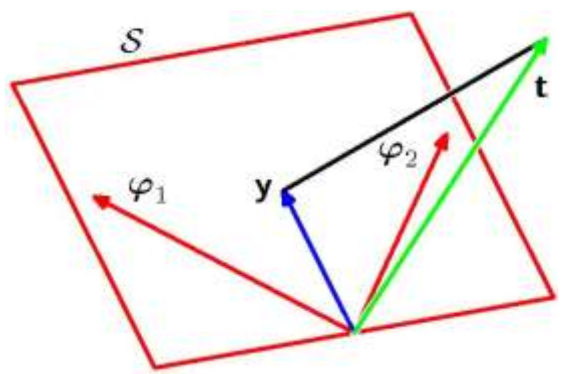
※ただし、基底関数の数Mは、データ点Nより小さい

二乗和誤差

$$\{\mathbf{t} - \mathbf{w}^T\phi(\mathbf{x}_n)\}^2 = (\mathbf{t} - \mathbf{y})^2$$

は**t**と**y**の「距離の二乗」

最尤推定解$\mathbf{w}_{ML}$を求めることは,
線形部分空間Sにあるベクトルの中で,
最も**t**と近いベクトルを求めること.



⇒Yはtの線形部分空間Sへの正射影

ペナルティ項を考えるその前に･･･②

学習のアルゴリズムについて考えよう

最尤推定のように一度にすべての訓練データ集合を処理する方法をバッチ手法という

バッチ手法はデータ集合が大規模だと計算に時間が掛かる問題がある

データ集合が大規模である場合は、データ点を一度に1つだけ用いてモデルのパラメータ順次更新していく逐次学習アルゴリズムを使うとよい

逐次学習のアルゴリズムは誤差関数を$E = \sum_n E_n$とすると

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n$$

これを繰り返してwを求める

誤差関数に正則化項を足して、最小化しよう

誤差関数

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{\mathbf{t} - \mathbf{w}^T\phi(\mathbf{x}_n)\}^2$$

に正則化項を加えた

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

を最小化する.

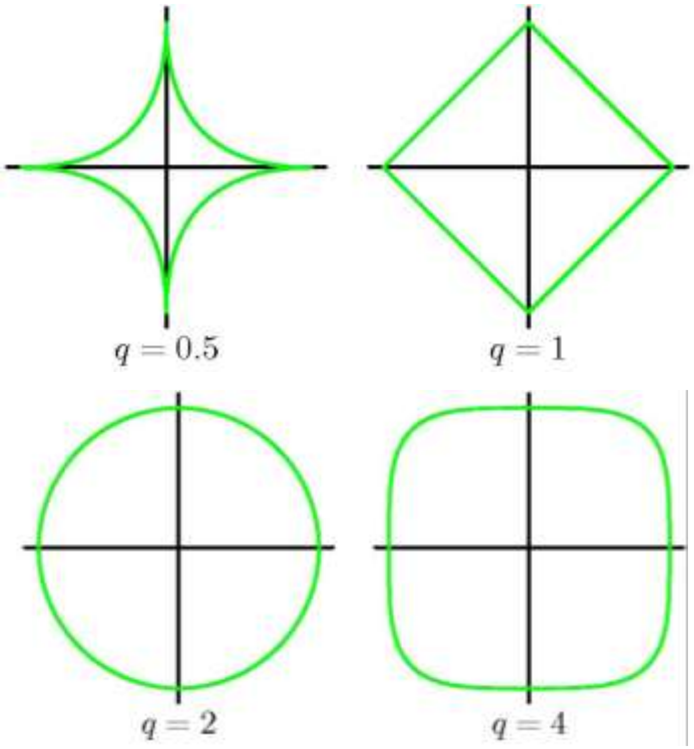
■ 正則化項の例

単純形: $\frac{1}{2}\mathbf{w}^T\mathbf{w}$

一般形: $\frac{1}{2}\sum_{j=1}^{M}|w_j|^q$

q=1のときlasso

例）様々なqに対する正則化項の等高線表示



誤差関数を最小化するwを求める

正則化項が単純系の場合、誤差関数全体は

$$\frac{1}{2}\sum_{n=1}^{N}\{\mathbf{t}-\mathbf{w}^T\phi(\mathbf{x}_n)\}^2+\frac{1}{2}\mathbf{w}^T\mathbf{w}$$

これを最大にするwを求めると

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

λの値は経験的に決めても良いのだが、バイアス・バリアンスというものを考慮に入れて決めることもできる

バイアス・バリアンスを考える前に、出力変数が多次元である場合を考える

多次元の場合の線形回帰モデルは以下のようになる

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{W}^T \boldsymbol{\varphi}(\mathbf{x})$$

ここで$\mathbf{y}$はK次元ベクトル、$\mathbf{W}$はM×K次元のパラメータ
$\mathbf{T} = (\boldsymbol{t_1}, \dots, \boldsymbol{t_N})^T$をN×K次元の行列で、n番目の行が$\boldsymbol{t}_\mathrm{n}{}^\mathrm{T}$となる行列だとする
このとき対数尤度関数を最大にする$\mathbf{W}$は

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

以後は単純化のため、1次元の目標変数tのみを考える

バイアスーバリアンス分解（1）

- 損失関数の予測値（条件付き期待値）

$$h(\mathbf{x}) = \mathbf{E}[t \mid \mathbf{x}] = \int t p(t \mid \mathbf{x}) dt$$

- 期待二乗損失

$$\mathbf{E}[L] = \int \boxed{\{y(\mathbf{x}) - h(\mathbf{x})\}^2} p(\mathbf{x}) d\mathbf{x} + \int\int \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t) d\mathbf{x} dt$$

この項を最小化したい。無限のデータがあれば0にできるが…データは有限個

- データ集合の取り方を考慮

$$\begin{aligned}
&\{y(\mathbf{x};\mathcal{D}) - h(\mathbf{x})\}^2 \\
&= \{y(\mathbf{x};\mathcal{D}) - \mathrm{E}_{\mathcal{D}}[\mathrm{y}(\mathbf{x};\mathcal{D})] + \mathrm{E}_{\mathcal{D}}[\mathrm{y}(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 \\
&= \{y(\mathbf{x};\mathcal{D}) - \mathrm{E}_{\mathcal{D}}[\mathrm{y}(\mathbf{x};\mathcal{D})]\}^2 + \{\mathrm{E}_{\mathcal{D}}[\mathrm{y}(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 \\
&\quad + 2\{y(\mathbf{x};\mathcal{D}) - \mathrm{E}_{\mathcal{D}}[\mathrm{y}(\mathbf{x};\mathcal{D})]\}\{\mathrm{E}_{\mathcal{D}}[\mathrm{y}(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}
\end{aligned}$$

バイアスーバリアンス分解（2）

- 期待値を取ると

$$\mathrm{E}_{\mathcal{D}}[\{y(\mathbf{x};\mathcal{D})-h(\mathbf{x})\}^2]$$

$$=\underbrace{\{\mathrm{E}_{\mathcal{D}}[\mathrm{y}(\mathbf{x};\mathcal{D})]-h(\mathbf{x})\}^2}_{（バイアス）^2}+\underbrace{\mathrm{E}_{\mathcal{D}}[\{[y(\mathbf{x};\mathcal{D})]-\mathrm{E}_{\mathcal{D}}[\mathrm{y}(\mathbf{x};\mathcal{D})]\}^2}_{バリアンス}]$$

- **バイアス**:
  平均的な推定値と真の値との差を測る

- **バリアンス**:
  訓練サンプルの違いに由来するランダムさを表す

バイアスーバリアンス分解（3）

- もとの損失関数に戻すと

$$\mathbf{E}[L] = \int \{y(\mathbf{x}) - h(\mathbf{x})\}^2 p(\mathbf{x}) d\mathbf{x} + \iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t) d\mathbf{x} dt$$

$$= \int \boxed{\{\mathrm{E}_{\mathcal{D}}[\mathrm{y}(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2} p(\mathbf{x}) d\mathbf{x}$$

$$+ \int \boxed{\mathrm{E}_{\mathcal{D}}[\{[y(\mathbf{x};\mathcal{D})] - \mathrm{E}_{\mathcal{D}}[\mathrm{y}(\mathbf{x};\mathcal{D})]\}^2]} \quad p(\mathbf{x}) d\mathbf{x}$$

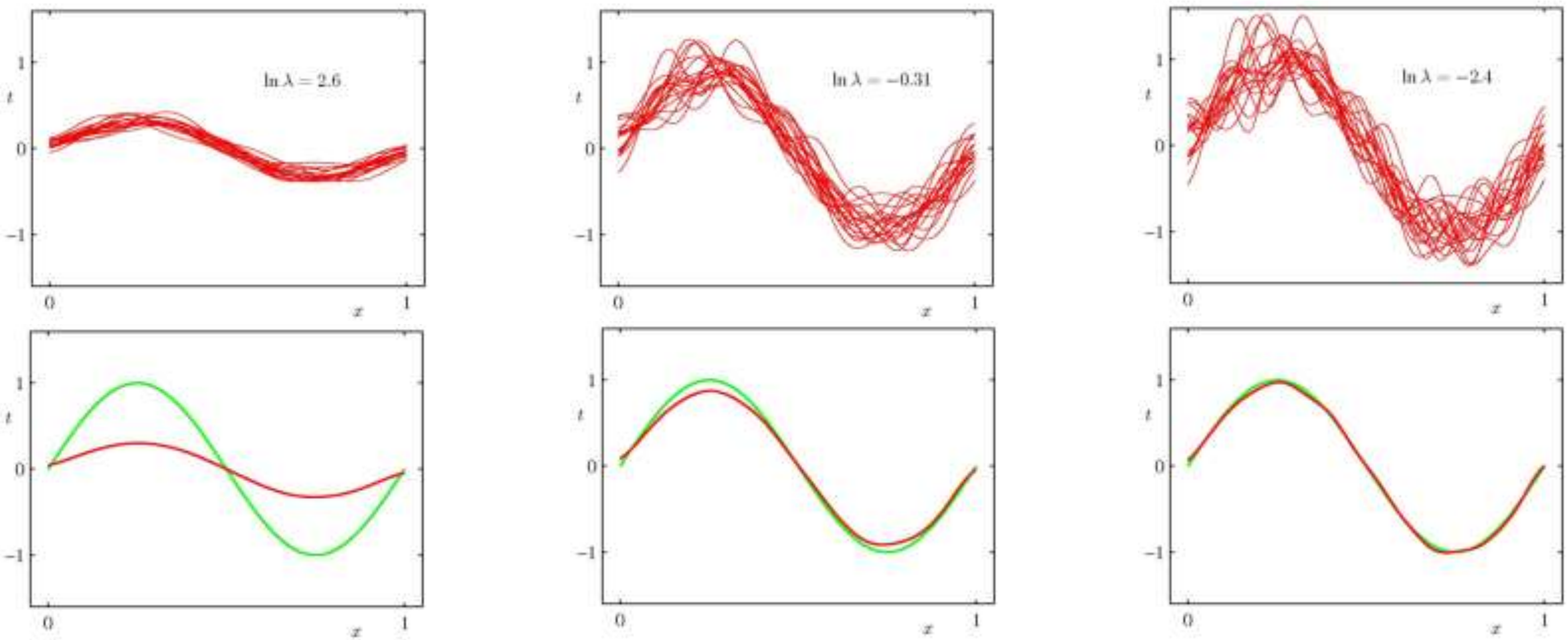
$$+ \boxed{\iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t) d\mathbf{x} dt}$$

$$= \boxed{(\text{バイアス})^2} + \boxed{\text{バリアンス}} + \boxed{\text{ノイズ}}$$

- バイアスとバリアンスをバランスよく小さくすることが必要

バイアスーバリアンス分解（4）

- 例）$h(x) = \sin(2\pi x)$
  - サンプル25点からなる100種類のデータ集合
  - 25個のガウス関数をフィット



バイアス大，バリアンス小　　　　バイアス小，バリアンス大

まとめ

訓練データからy(x,**w**)を求めて、新しい入力にも対応したい

何を求めれば良い？

パラメータ**w**、基底関数の数を求めたい

どうやって？

最尤推定で求める（頻度主義の考え）

データに対して基底関数が多いと過学習が起きる

正則化項を加えた誤差関数を最小にすることで過学習を抑える
なお、正則化項の係数の値はバイアス・バリアンスを考慮に入れて決めることもできる

ベイジアンで解く！

パラメータ$\mathbf{w}$の分布が知りたい

どうやって？

事前分布に訓練データを反映させて事後分布を求める

条件

事前分布と事後分布が同じ分布であること

条件を満たす事前分布はどんな形？

尤度関数の形に注目する

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

$$=\prod_{n=1}^{N}\frac{\beta^{\frac{1}{2}}}{\sqrt{2\pi}}\exp\{-\frac{1}{2}\beta(t_n-y(\mathbf{x}_n,\mathbf{w}))^2\}$$

指数にwの二次関数を持っている

このような形の場合、事前分布には正規分布を使うと上の条件を満たせることがわかっている。
また、計算を楽にするために、平均が0，共分散行列が単位行列の定数倍というとても扱いやすい正規分布を使うことにする

どうやって事後分布を求めるか？

ベイズの定理を使う

$$p(\mathbf{w}\mid\mathcal{D})=\frac{p(\mathcal{D}\mid\mathbf{w})p(\mathbf{w})}{\boxed{p(\mathcal{D})}}$$

定数

よって、事後分布は

$$\text{事後分布} \propto \text{尤度関数} \times \text{事前分布}$$

よって、事後分布は尤度関数と事前分布をかけて、
正規化すれば求められる

wの事後分布を求める

wの事前分布は

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|0, \alpha^{-1}\mathbf{I})$$

とすると、wの事後分布は

$$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathrm{m}_N, \mathrm{S}_N)$$

$$\mathrm{m}_N = \beta \mathrm{S}_N \Phi^T \mathbf{t}$$

$$\mathrm{S}_\mathrm{N}^{-1} = \alpha \mathbf{I} + \beta \Phi^T \Phi$$

事後分布の対数を取ると、二乗和誤差関数と二次正則化項

$$\ln p(\mathbf{w}|\mathbf{t}) = -\frac{\beta}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_n)\}^2 - \frac{\alpha}{2}\mathbf{w}^T\mathbf{w} + \text{定数}$$

したがって、この事後分布をwに関して最大化することは正則化項を加えた誤差関数を最小化することと等価である

ベイズ線形回帰の例をみてみよう

設定

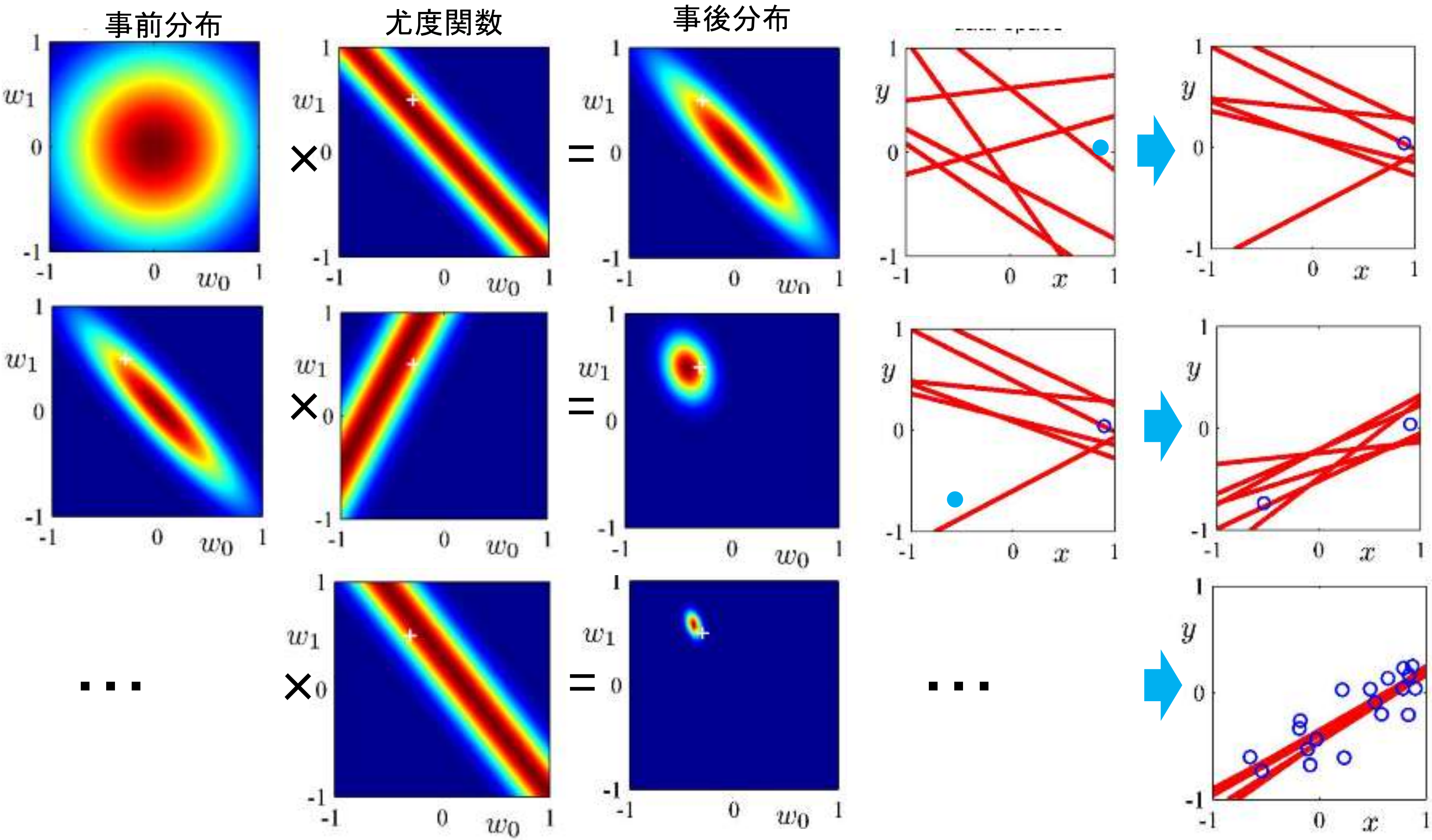
- 例）線形基底関数モデル $y(x, \mathbf{w}) = w_0 + w_1 x$

  関数 $f(x, \mathbf{a}) = a_0 + a_1 x \quad (a_0 = -0.3, a_1 = 0.5)$

  を復元する.

1. 上のデータから複数の訓練データを出す
2. 訓練データから一つ適当に取り出す
3. 取り出したデータから尤度関数を求める
4. 尤度関数と事前分布をかけて，パラメータの事後分布を求める
5. パラメータの事後分布から適当に取り出し，関数を推定する.
6. 訓練データからまた一つ取り出す
7. 3～6を繰り返す

ベイズ線形回帰の例(2)



予測分布を求める

回帰の目的である関数を求めるのではなく、新たな入力が与えられた時の目標値を予測することを考える

そこで、予測分布を求める

求めた事後分布を用いて、予測事後分布を求める

$$p(t|\mathbf{t},\alpha,\beta)=\int p(t|\mathbf{w},\beta)p(\mathbf{w}|\mathbf{t},\alpha,\beta)d\mathbf{w}$$

ただし $p(t|\mathbf{w},\beta)=\mathcal{N}(t|y(\mathbf{x},\mathbf{w}),\beta^{-1})$

$$p(\mathbf{w}|\mathbf{t},\alpha,\beta)=\mathcal{N}(\mathbf{w}|\mathrm{m}_N,\mathrm{S}_N)$$

- 結局

$$p(t|\mathbf{x},\mathbf{t},\alpha,\beta)=\mathcal{N}(\mathrm{t}|\mathbf{m}_{\mathrm{N}}^{\mathrm{T}}\phi(\mathbf{x}),\sigma_N^2(\mathbf{x}))$$

ただし $\sigma_N^2(\mathbf{x})=\boxed{\frac{1}{\beta}}+\boxed{\phi(\mathbf{x})^T\mathbf{S}_N\phi(\mathbf{x})}$

Wに関する不確かさ

データに含まれる
ノイズ

予測分布は分散が小さくなることが証明されている

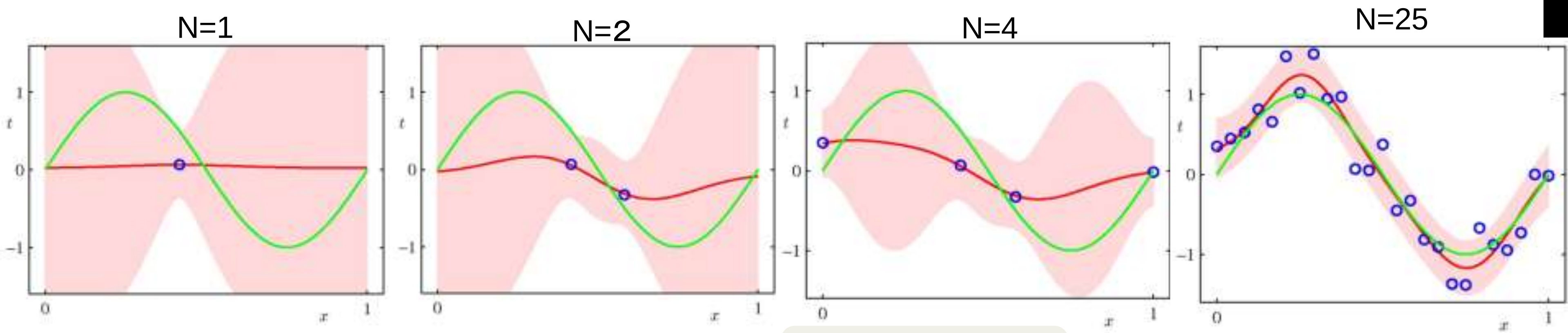
分散が小さくなる証明

$$
\begin{aligned}
\sigma_{N+1}^2(\mathbf{x}) &= \beta^{-1} + \boldsymbol{\varphi}(\mathbf{x})^T \mathbf{S}_{N+1} \boldsymbol{\varphi}(\mathbf{x}) \\
&= \beta^{-1} + \boldsymbol{\varphi}(\mathbf{x})^T (\mathbf{S}_N^{-1} + \beta \boldsymbol{\varphi}(\mathbf{x}) \boldsymbol{\varphi}(\mathbf{x})^T)^{-1} \boldsymbol{\varphi}(\mathbf{x}) \\
&= \beta^{-1} + \boldsymbol{\varphi}(\mathbf{x})^T (\mathbf{S}_N - \frac{\beta \mathbf{S}_N \boldsymbol{\varphi}(\mathbf{x}) \boldsymbol{\varphi}(\mathbf{x})^T \mathbf{S}_N}{1 + \beta \boldsymbol{\varphi}(\mathbf{x})^T \mathbf{S}_N \boldsymbol{\varphi}(\mathbf{x})}) \boldsymbol{\varphi}(\mathbf{x}) \\
&= \beta^{-1} + \boldsymbol{\varphi}(\mathbf{x})^T \mathbf{S}_N \boldsymbol{\varphi}(\mathbf{x}) - \beta \boldsymbol{\varphi}(\mathbf{x})^T \frac{\mathbf{S}_N \boldsymbol{\varphi}(\mathbf{x}) \boldsymbol{\varphi}(\mathbf{x})^T \mathbf{S}_N}{1 + \beta \boldsymbol{\varphi}(\mathbf{x})^T \mathbf{S}_N \boldsymbol{\varphi}(\mathbf{x})} \boldsymbol{\varphi}(\mathbf{x}) \\
&= \sigma_N^2(\mathbf{x}) - \beta \frac{(\boldsymbol{\varphi}(\mathbf{x})^T \mathbf{S}_N \boldsymbol{\varphi}(\mathbf{x}))(\boldsymbol{\varphi}(\mathbf{x})^T \mathbf{S}_N \boldsymbol{\varphi}(\mathbf{x}))}{1 + \beta \boldsymbol{\varphi}(\mathbf{x})^T \mathbf{S}_N \boldsymbol{\varphi}(\mathbf{x})} \\
&\leq \sigma_N^2(\mathbf{x})
\end{aligned}
$$

よって、データがあれば有るほど予測値のばらつきを抑える事ができる

予測分布は訓練データを得るごとに、分散が小さくなっていくことの例を示す

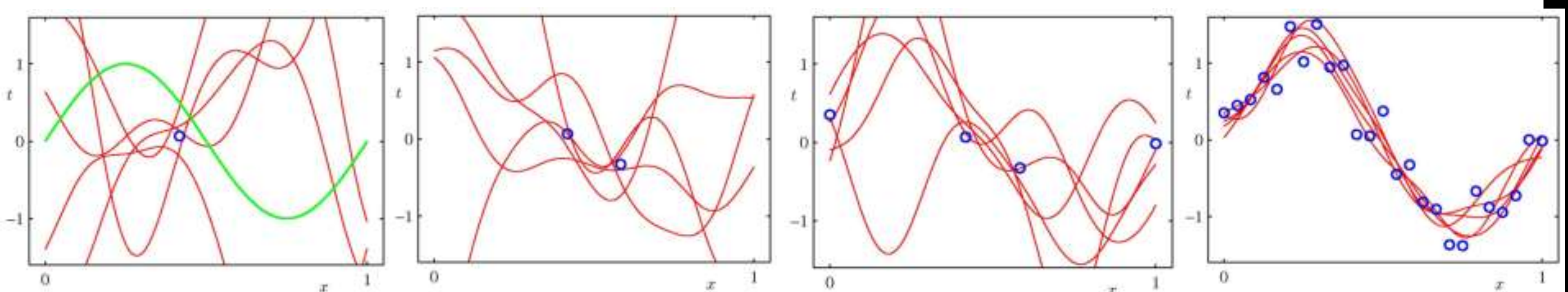
- 例）ガウス基底関数結合モデルの $\sin(2\pi x)$ へのあてはめ



ガウス予測分布の平均

+-標準偏差

wの事後分布から選んでプロットしたy(x, **w**)



等価カーネル（1）

- 訓練データの目標値だけから予測する
- 線形基底関数モデルに対して
  事後分布の平均解を導入

$$y(\mathbf{x}, \mathrm{m}_N) = \mathrm{m}_N^T \phi(\mathbf{x}) = \beta \phi(\mathbf{x})^T \mathbf{S}_N \Phi^T \mathbf{t} = \sum_{n=1}^{N} \beta \phi(\mathbf{x})^T \mathbf{S}_N \phi(x_n) t_n$$

$$\mathrm{m}_N = \beta \mathrm{S}_N \Phi^T \mathbf{t}$$

$$\mathrm{S}_{\mathrm{N}}^{-1} = \alpha I + \beta \Phi^T \Phi$$

- つまり，訓練データの目標値t_nの線形結合

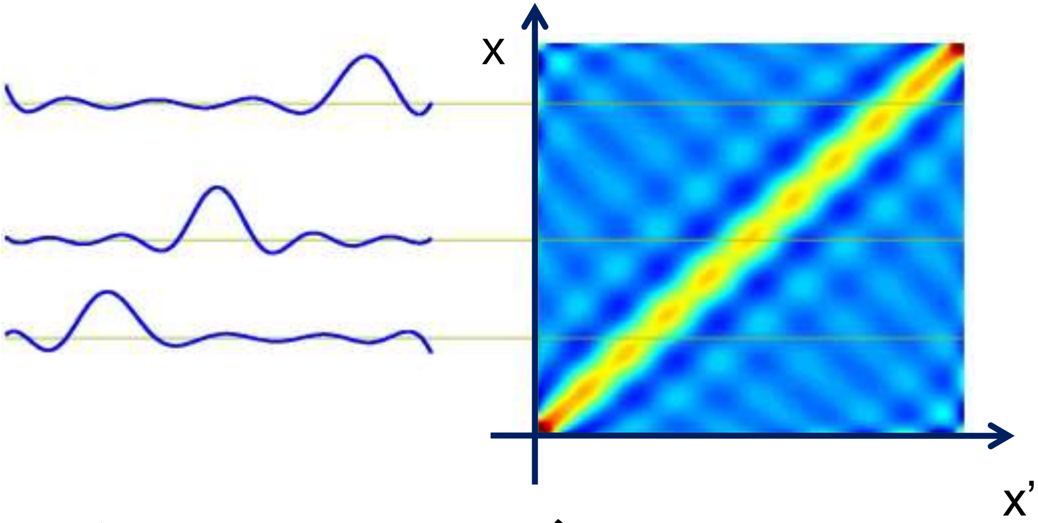
$$\boxed{y(\mathbf{x}, \mathrm{m}_N) = \sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n) t_n}$$

$$k(\mathbf{x}, \mathbf{x}') = \beta \phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x}')$$

平滑化行列または等価カーネル

等価カーネル（2）

- ガウス基底関数に対するk(x,x')をプロット



⇒xに近いx'を大きく重みづけ

まとめ

訓練データからy(x,**w**)を求めて、新しい入力にも対応したい

何を求めれば良い？

パラメータ**w**、基底関数の数を求めたい

どうやって？

最尤推定で求める（頻度主義の考え）

事後分布を求める（ベイジアンの考え）

データに対して基底関数が多いと過学習が起きる

新たな入力に対する目標値を求めたいことだけを考える

正則化項を加えた誤差関数を最小にすることで過学習を抑える

予測分布を求める