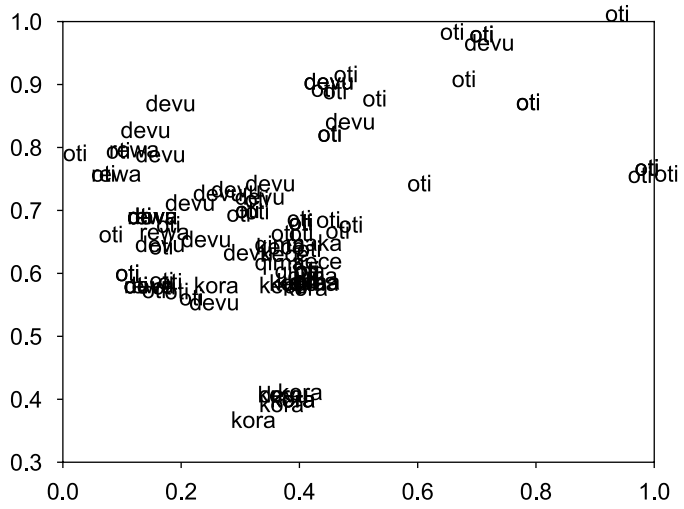
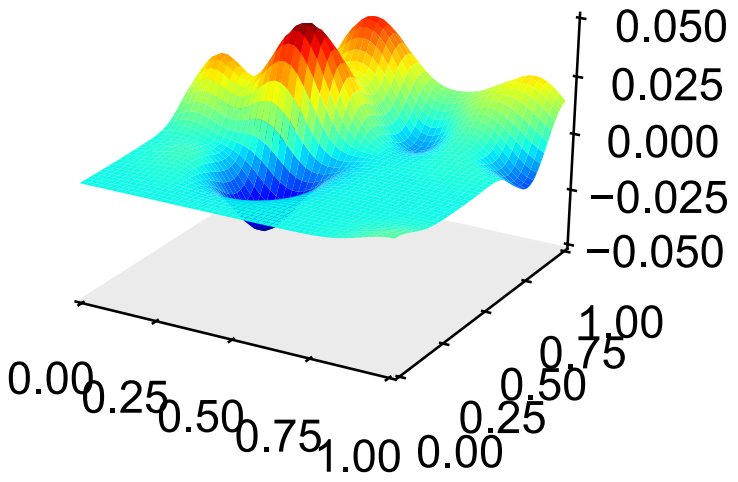
ガウス過程と自然言語処理





持橋大地

統計数理研究所

daichi@ism.ac.jp

言語処理学会2021チュートリアル T1

2021-3-15 (月)

小倉・北九州国際会議場

自己紹介

- 統計数理研究所 数理・推論研究系 准教授／
  総合研究大学院大学 統計科学専攻
- 東大基礎科→NAIST 松本研、ATR音声研、NTT CS研を経て、2011年から現職
- 専門: 自然言語処理 (特に言語モデル)、機械学習

立川・統数研

前回のチュートリアル (2006年3月)

修正版, 2006.3.13)

*Topic*に基づく
統計的言語モデルの最前線

— PLSIからHDPまで —

山本幹雄
筑波大学)

持橋大地
(ATR)

URL= http://www.mibel.cs.tsukuba.ac.jp/~myama/pdf/topic2006.pdf

言語処理学会第12回年次大会チュートリアル, 2006.3.13

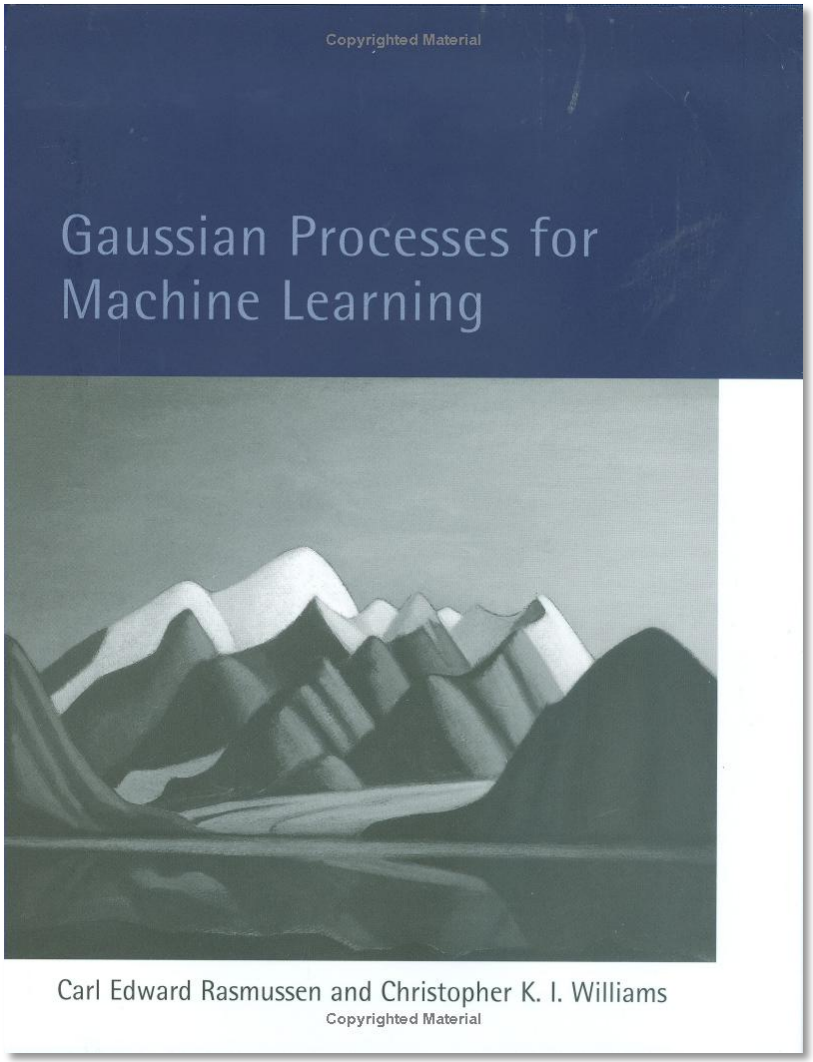
今日の概要

- はじめに: ガウス過程回帰とは何か
- 線形回帰モデル
- ガウス過程とガウス過程回帰
- カーネル関数の学習
- 深層学習との関係
- 自然言語処理への様々な応用

教科書「ガウス過程と機械学習」

- 講談社機械学習プロフェッショナルシリーズ(MLP), 2019/3/9発売
  - 持橋大地・大羽成征著
  - 現在、レビュー38件
- 線形回帰モデルの非常にやさしい導入から入っています
- 確率過程としての話ではなく、統計の道具としての意味と使い方の話

教科書 (GPML)



- “Gaussian Processes for Machine Learning” by Carl Rasmussen & Chris Williams
- 中級者以上向け
- 数学的な詳細やカーネル設計などについて知りたい場合はこちら
- PDFがフリーでダウンロード可能 http://www.gaussianprocess.org/gpml/

ACL 2014 Tutorial on GP

- 数式が多く、今回の方がかなりわかりやすいはず
- CohnらのGPを使った研究を後半で説明

Gaussian Processes for
Natural Language Processing

http://goo.gl/18heUk

Trevor Cohn[1] Daniel Preoţiuc-Pietro[2] Neil Lawrence[2]

Computing and Information Systems[1] Department of Computer Science[2]

THE UNIVERSITY OF MELBOURNE

The University Of Sheffield.

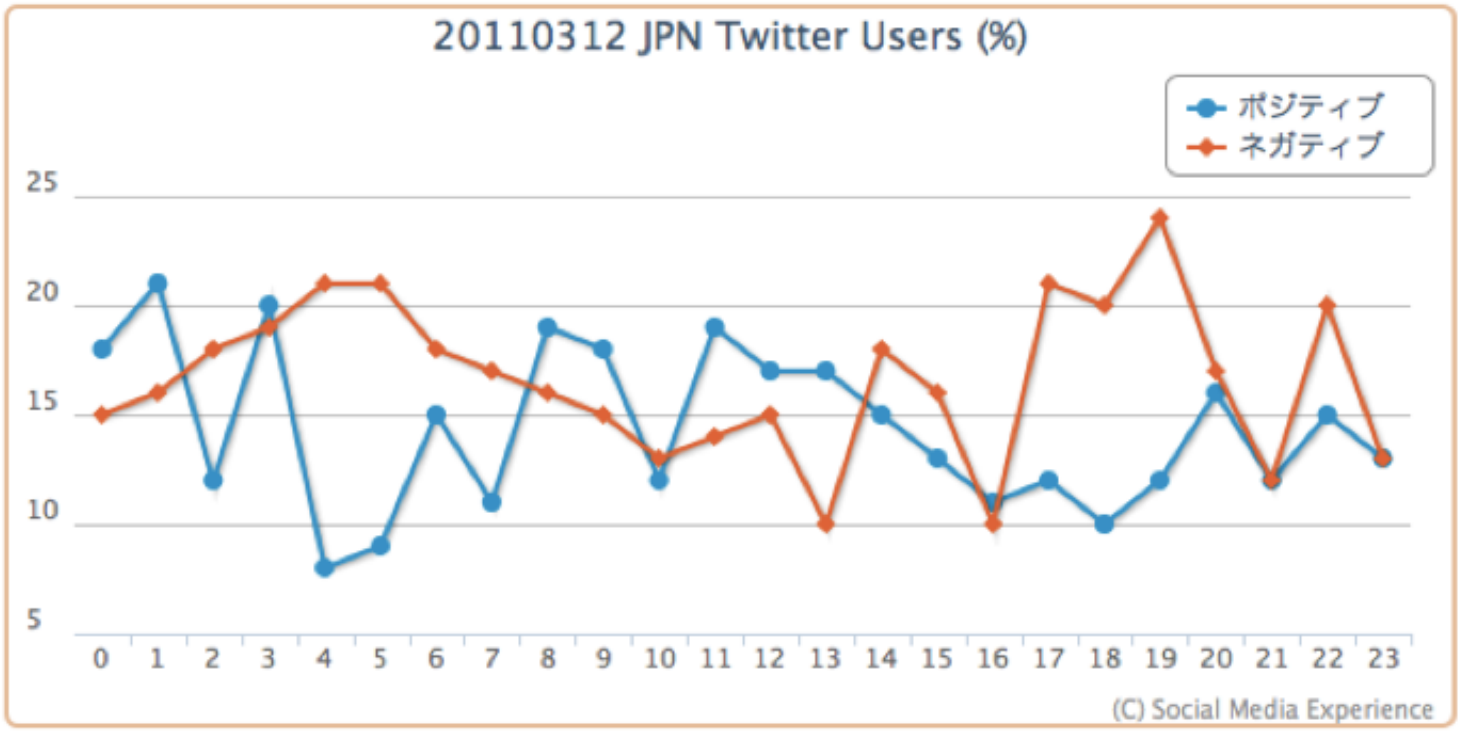
ACL 2014 Tutorial, 22 June 2014

(Special thanks also to Daniel Beck)

なぜガウス過程？

- 自然言語処理でも、連続値を扱う機会が増えている／カテゴリに分類すれば終わりではない
  - 単語ベクトルの座標
  - 時間データ
  - 地理データ
  - 物理データ (ロボットや車の動作など)
  - 価格、得点、評価値など
- ニューラルネットにただ突っ込むだけでは、あまりにも貧しい (きちんとした道具が必要)

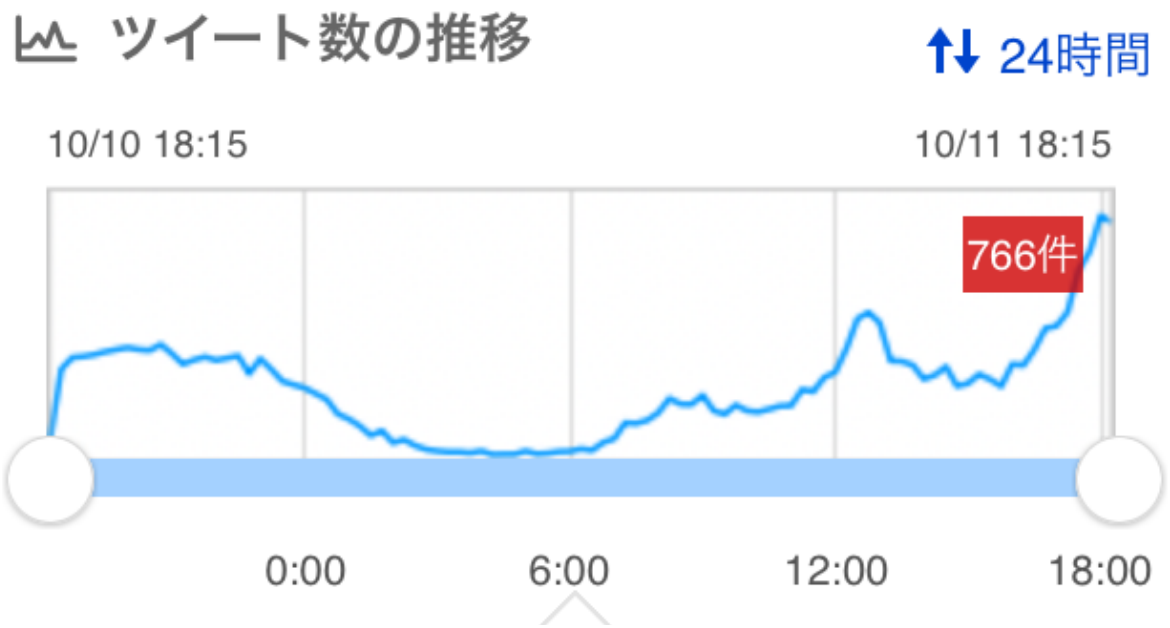
例) 時間によるツイート内容の推移



http://socialmediaexperience.jp/3192 より引用

- 東日本大震災後、2011/3/12のツイートの時間変化
  - 簡単な分布では表現できない！

例) おまけ・台風コロッケ

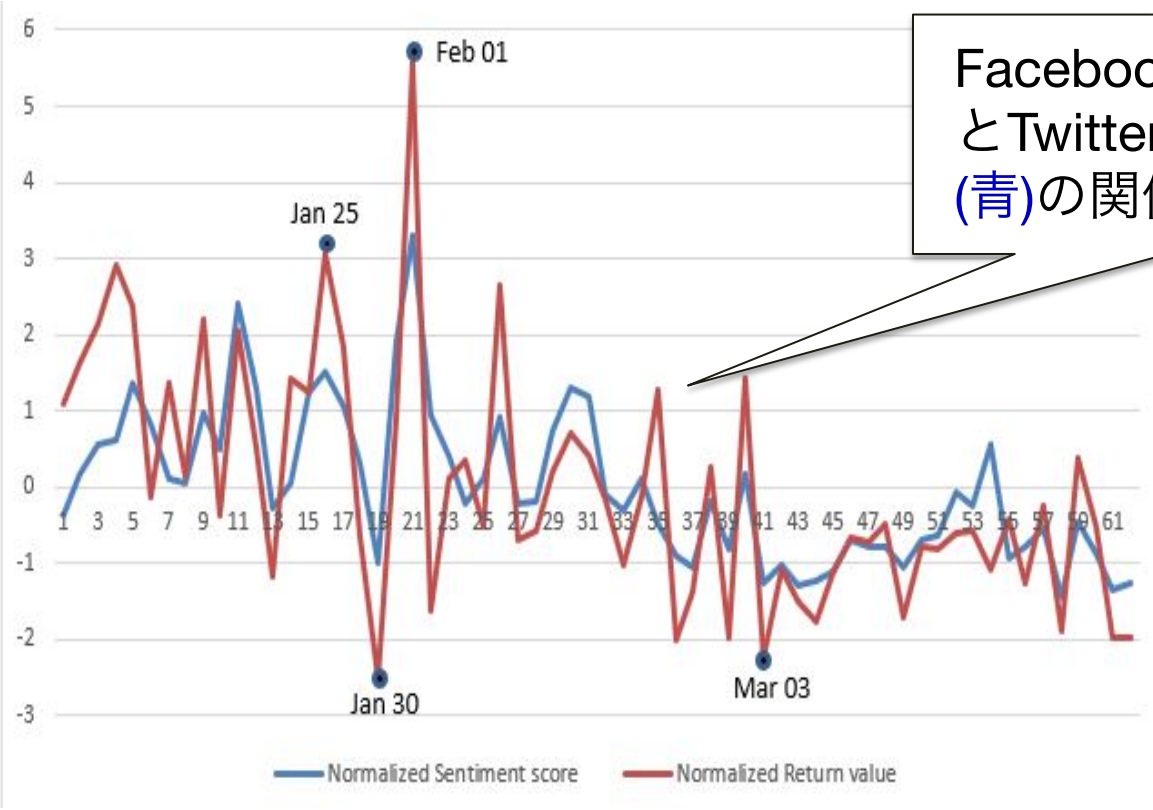


- 2019年10月11日の台風時の「コロッケ」のツイート頻度と時間の関係
- 非常に滑らかな関数！

例) Twitterの座標と言葉

- 2009年スーパーボウル時のツイートの単語頻度と座標 (New York Times)

例) 株価とツイートの感情分析

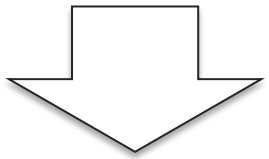


https://www.aclweb.org/anthology/W18-3102/ より引用

- “Causality Analysis of Twitter Sentiments and Stock Market Returns”, ACL 2018 WS in Economics and Natural Language Processing より

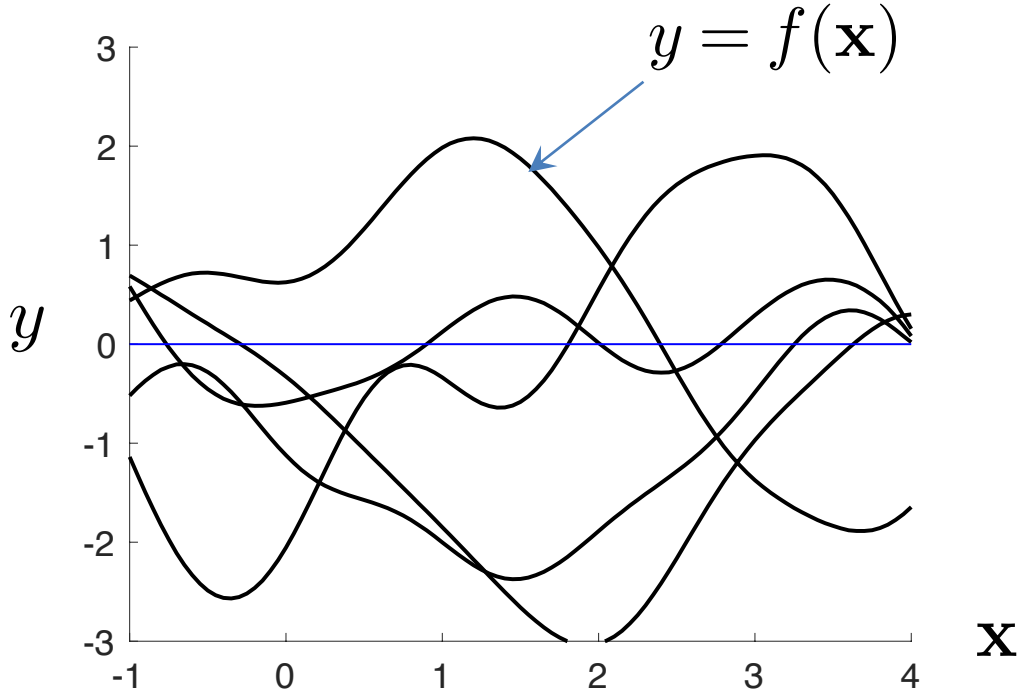
連続値への回帰問題

- これらは、入力 $\mathbf{x} \mapsto$ 出力 $y \in \mathbb{R}$ (連続値) を予測する問題 **(回帰問題、regression)**
  - 分類器では対応できない
  - 通常のガウス分布など単純な分布も使えない
- モデルがないと、無理矢理ニューラルネットに入れても解けない：動作がまったく保証されない



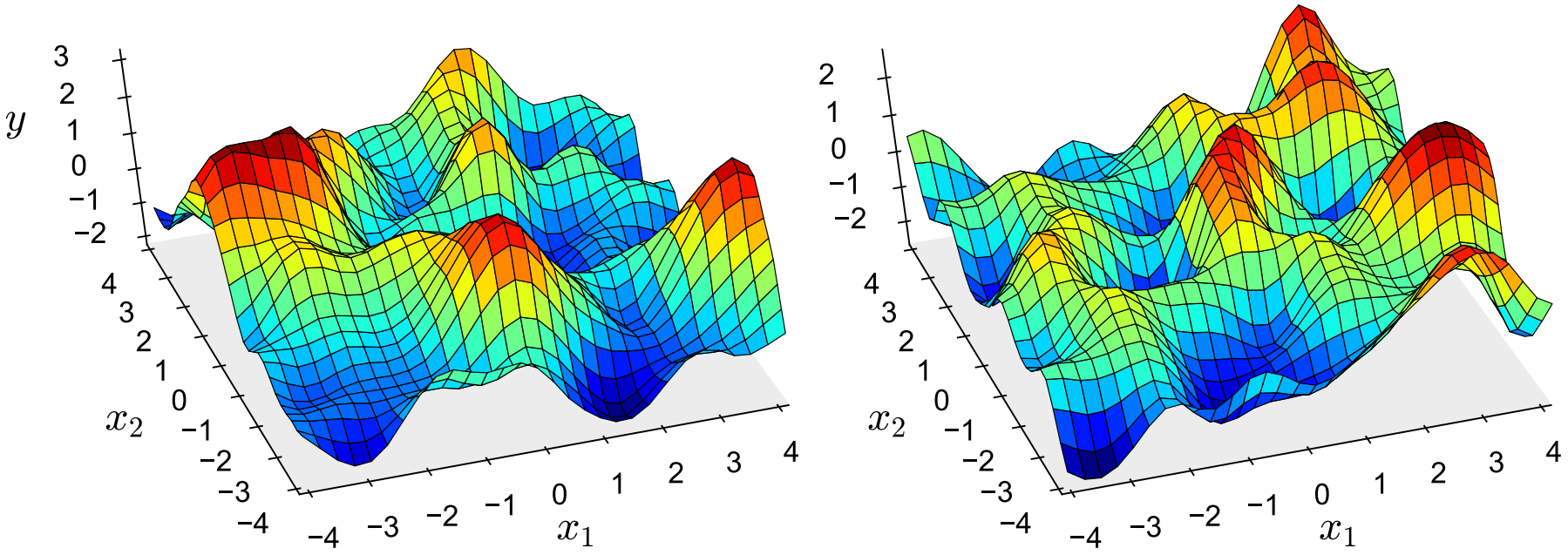
道具を増やす必要がある

ガウス過程とは?



- 非常に柔軟な回帰関数 $f : \mathbf{x} \mapsto y$ を生成する確率モデル (関数の確率分布)
- カーネル関数 $k(\mathbf{x}, \mathbf{x}')$ によって様々な関数が生成できる (ベイズ的なカーネル法)

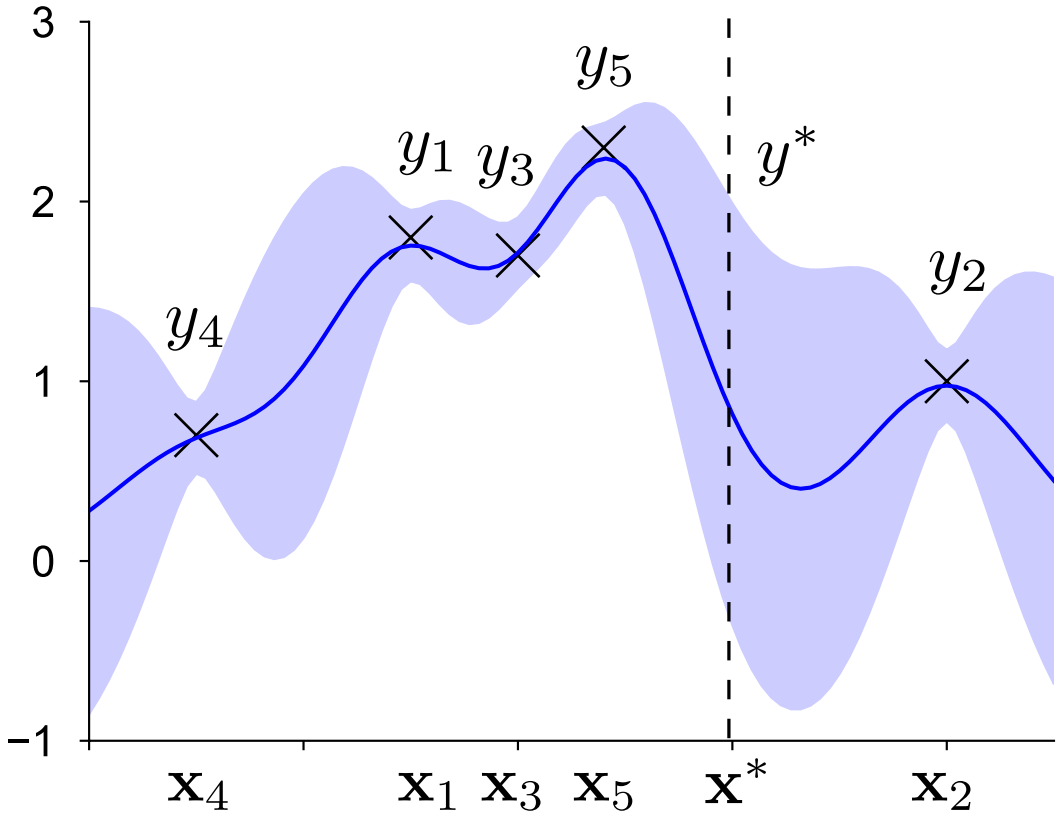
ガウス過程とは? (2)



- 入力が2次元の場合のガウス過程からのサンプル
  ＝ランダムな連続曲面
- 入力xがもっと高次元な場合も同様のイメージ

ガウス過程とは? (3)

- 関数のベイズ推定 : データが与えられると、関数の事後分布が得られる

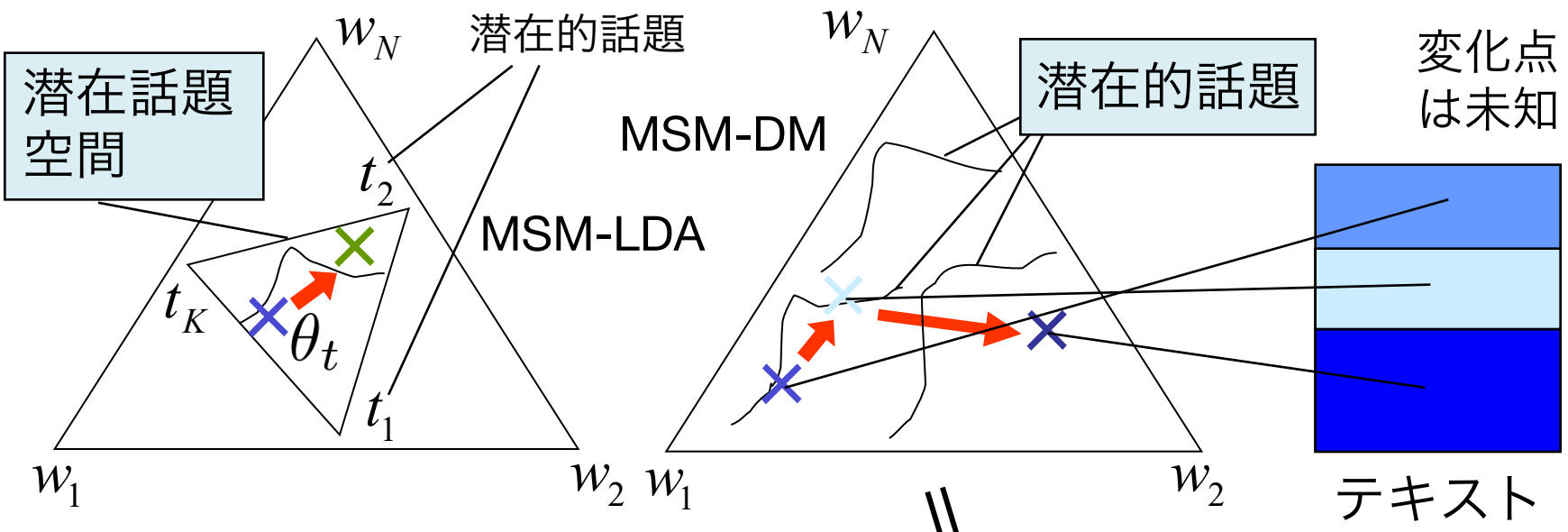


- 青線は期待値
- データのない場所は分散が大きい
- 通常の最適化では分散は表現できない

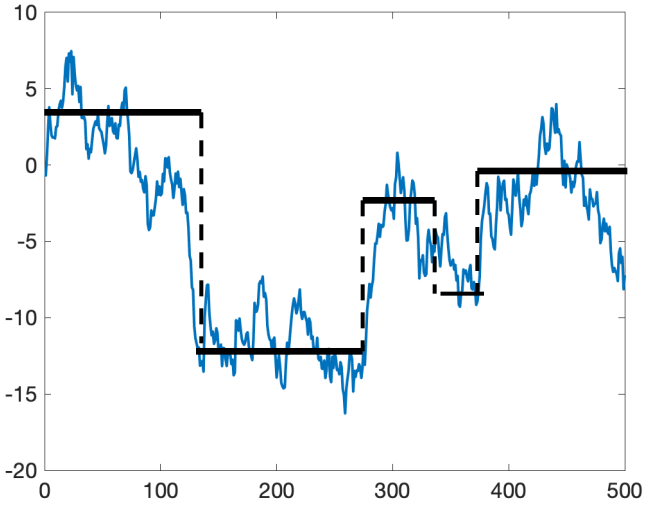
ガウス過程とは? (4)

- 柔軟な回帰関数を使える確率モデルなので、全体を見通しのよい統計モデルとして定義できる
  - 統計的な振る舞いが保証されている ($\leftrightarrow$ NN)
  - ハイパーパラメータも同時に学習できる
- カーネル法なので、多くの入力 $\mathbf{x}$ に対して自然に定義できる
  - 高次元の入力
  - 文字列、グラフ、木、確率モデル、..
- ニューラルネットは、素子数$\rightarrow \infty$でガウス過程になる (Neal 1996)

私の博士論文 (“Context as Filtering”, NIPS 2005)



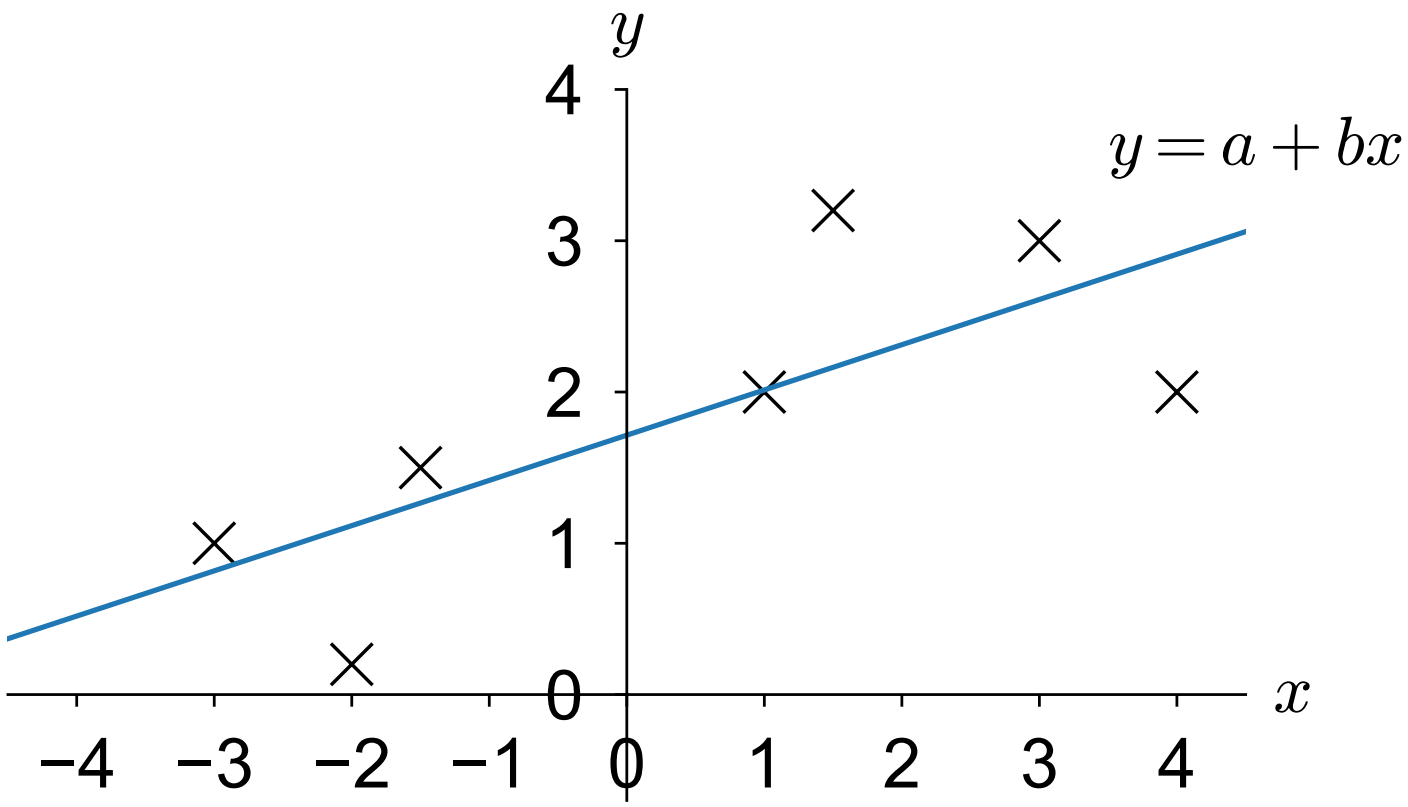
- 潜在的な文脈の変化を Particle Filterで追跡する問題
- ガウス過程はまだ知られていなかった



線形回帰モデル

単回帰モデル (simple regression)

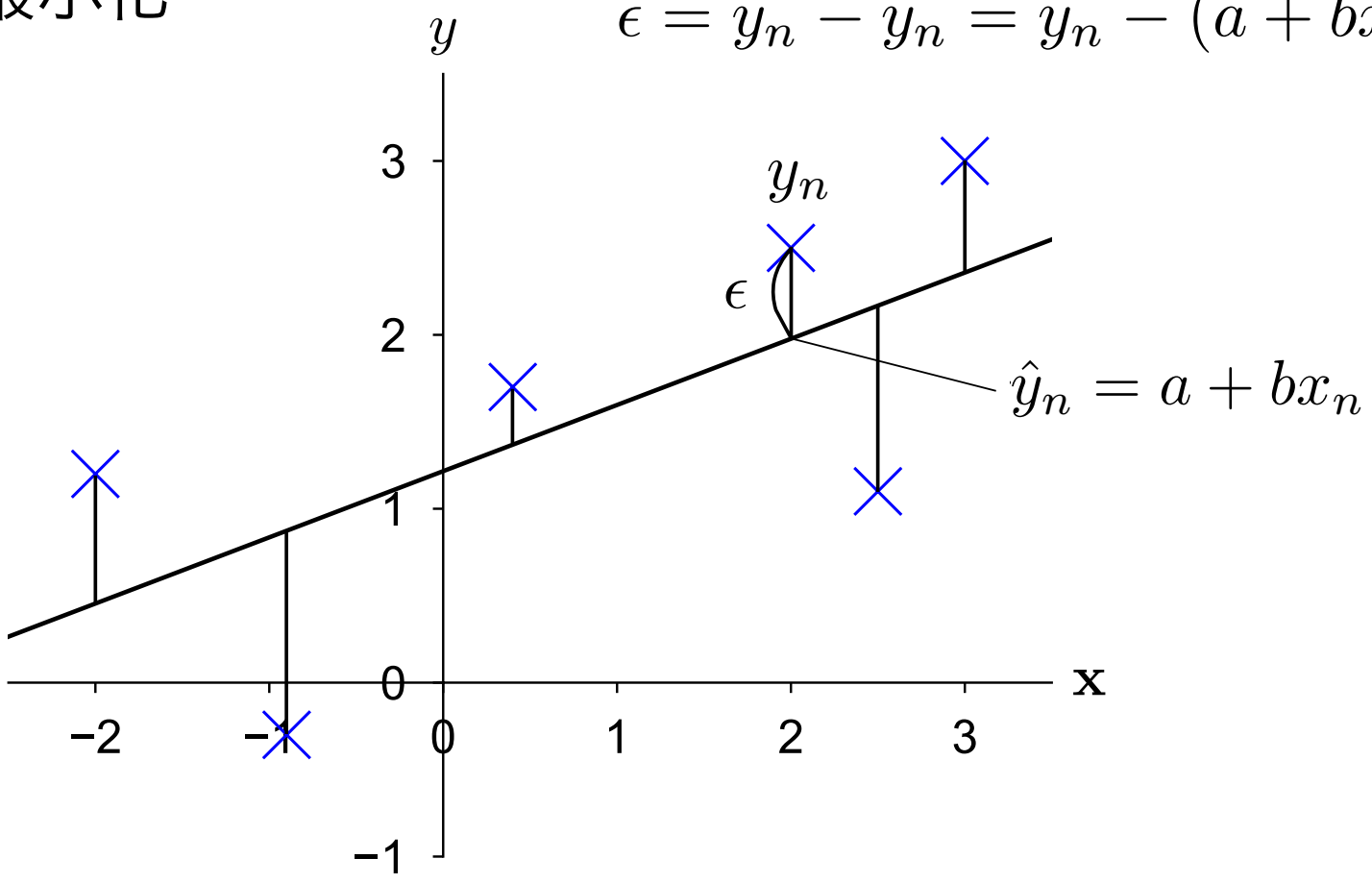
- 最も単純な回帰： $y = a + bx$
- aとbをどうやって決める?



誤差の最小化

- 実際の値 y_n と予測値 $\hat{y}_n = a + b\,x_n$ の誤差 ϵ を最小化

$$\epsilon = y_n - \hat{y}_n = y_n - (a + bx_n)$$



単回帰モデル (2)

- データ $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$ があったとする.
- 各 x_n に対する予測値 $\hat{y}_n$ は、一次式

$$\hat{y}_n = a + bx_n$$

- 観測値との差は

$$y_n - \hat{y}_n = y_n - (a + bx_n)$$

  - これを最小にしたい！

単回帰モデル (3)

- *n*=1,2,...,*N* について、
  誤差＝ $y_n - \hat{y}_n$ → 誤差の総和を最小にしたい
- 誤差は負のこともあるので、二乗した二乗誤差を最小化 (最小二乗法)：

$$E = \sum_{n=1}^{N} (y_n - \hat{y}_n)^2 = \sum_{n=1}^{N} (y_n - (a + bx_n))^2$$

  を最小にする a,b を求める

単回帰モデル (4)

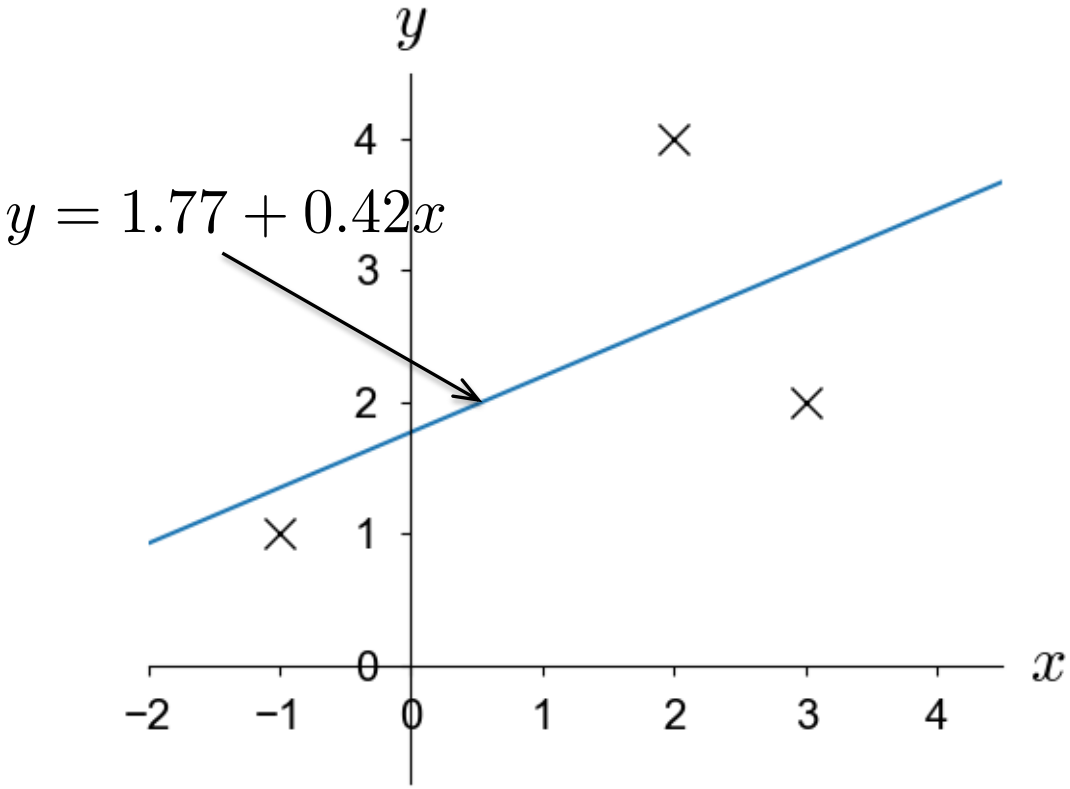
- Eの極小点ではa,bについての偏微分は0になるので、

$$\frac{\partial E}{\partial a} = \frac{\partial}{\partial a}\sum_{n=1}^{N}(y_n - (a + bx_n))^2$$
$$= \frac{\partial}{\partial a}\sum_{n=1}^{N}(y_n^2 + a^2 + b^2x_n^2 - 2ay_n - 2abx_n + 2bx_ny_n) = 0$$
$$\frac{\partial E}{\partial b} = \frac{\partial}{\partial b}\sum_{n=1}^{N}(y_n - (a + bx_n))^2$$
$$= \frac{\partial}{\partial b}\sum_{n=1}^{N}(y_n^2 + a^2 + b^2x_n^2 - 2ay_n - 2abx_n + 2bx_ny_n) = 0$$

- これを解いて、

$$a = \frac{\sum_n x_n^2 \sum_n y_n - \sum_n x_n \sum_n x_n y_n}{N\sum_n x_n^2 - (\sum_n x_n)^2}$$
$$b = \frac{N\sum_n x_n y_n - \sum_n x_n \sum_n y_n}{N\sum_n x_n^2 - (\sum_n x_n)^2}$$

単回帰モデルの計算例



一番単純な場合：
データD=
{(3,2),(2,4),(-1,1)}

$$\sum_{n=1}^{3} x_n = 3 + 2 - 1 = 4$$

$$\sum_{n=1}^{3} y_n = 2 + 4 + 1 = 7$$

$$\sum_{n=1}^{3} x_n^2 = 9 + 4 + 1 = 14$$

$$\sum_{n=1}^{3} x_n y_n = 3 \cdot 2 + 2 \cdot 4 + (-1) \cdot 1 = 13$$

- 公式に代入して、
  a = 1.77, b= 0.42

重回帰モデル

- 入力xが多次元なら? → 重回帰 (multiple regression)

$$\mathbf{x} = (x_1, x_2, \cdots, x_D)^T \text{ のとき、}$$

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_D x_D$$

- 二乗誤差は、

$$(y - \hat{y})^2 = (y - (w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_D x_D))^2$$

- これを最小化
  → $E = \sum_{n=1}^{N} (y_n - \hat{y}_n)^2$ を $w_0, w_1, \cdots, w_D$ について微分して0とおき、連立方程式を解けばよい.

もっと見通しよく!

- x を新しく $\mathbf{x} = (1, x_1, x_2, \cdots, x_D)$ 、
  重みベクトルを $\mathbf{w} = (w_0, w_1, w_2, \cdots, w_D)$ と表せば、

$$
\begin{aligned}
\hat{y} &= w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_D x_D \\
&= (w_0, w_1, w_2, \cdots, w_D) \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_D \end{pmatrix} \\
&= \mathbf{w}^T \mathbf{x}
\end{aligned}
$$

もっと見通しよく! (2)

- よって、n=1,2,...,N について縦に並べれば、

$$\begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_N \end{pmatrix} = \begin{pmatrix} \mathbf{w}^T\mathbf{x}_1 \\ \mathbf{w}^T\mathbf{x}_2 \\ \vdots \\ \mathbf{w}^T\mathbf{x}_N \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{pmatrix} \mathbf{w}$$

計画行列という

- つまり、

$$\underbrace{\begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_N \end{pmatrix}}_{\hat{\mathbf{y}}} = \underbrace{\begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1D} \\ 1 & x_{21} & x_{22} & \cdots & x_{2D} \\ \vdots & & & & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{ND} \end{pmatrix}}_{\mathbf{X}} \underbrace{\begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_D \end{pmatrix}}_{\mathbf{w}}$$

$\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}$ と書ける!

行列・ベクトル表現

$$E = \sum_{n=1}^{N} (y_n - \hat{y}_n)^2 = (y_1 - \hat{y}_1, \cdots, y_N - \hat{y}_N) \begin{pmatrix} y_1 - \hat{y}_1 \\ \vdots \\ y_N - \hat{y}_N \end{pmatrix}$$

- なので、

$$\begin{aligned} E &= (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w}) \\ &= \mathbf{y}^T (\mathbf{y} - \mathbf{X}\mathbf{w}) - (\mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w}) \\ &= \mathbf{y}^T \mathbf{y} - 2\mathbf{w}^T (\mathbf{X}^T \mathbf{y}) + \mathbf{w}^T \mathbf{X}^T \mathbf{X}\mathbf{w} \end{aligned}$$

重回帰モデルの解

$$E = \mathbf{y}^T\mathbf{y} - 2\mathbf{w}^T(\mathbf{X}^T\mathbf{y}) + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}$$

- を $\mathbf{w}$ で微分して、

$$\frac{\partial E}{\partial \mathbf{w}} = -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{0}$$

- よって

$$\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y} \qquad \text{(正規方程式)}$$

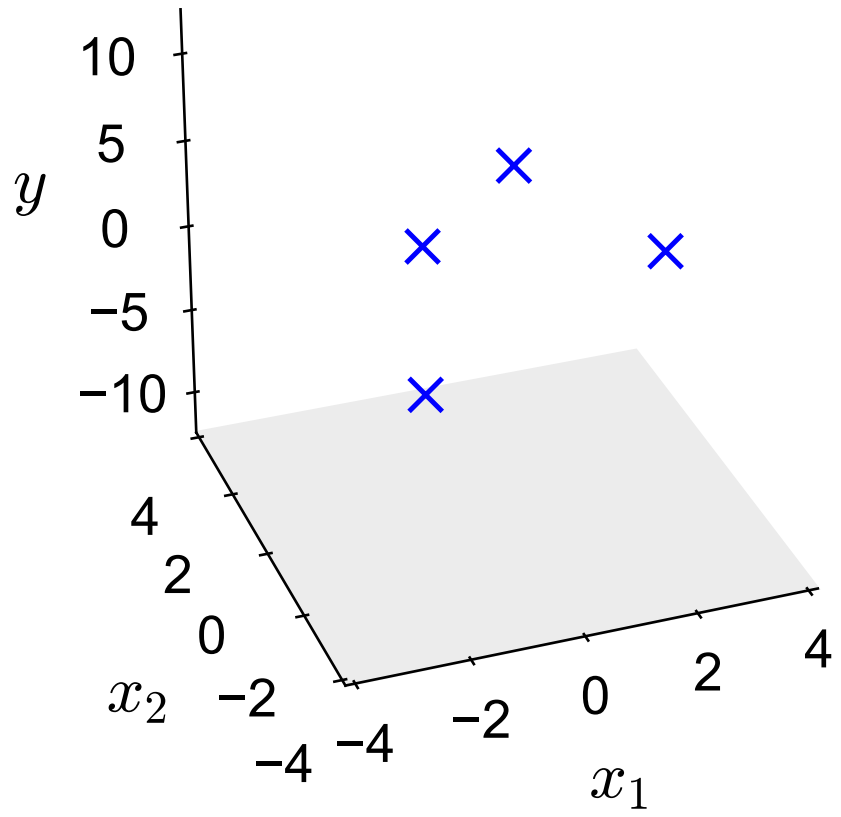
$$\therefore \quad \boxed{\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}\,.}$$

重回帰モデルの解

重回帰モデルの計算例

- データが下のとき、

$$\mathcal{D} = \{((1,2),4),((-1,1),2),((3,0),1),((-2,-2),-1)\}$$



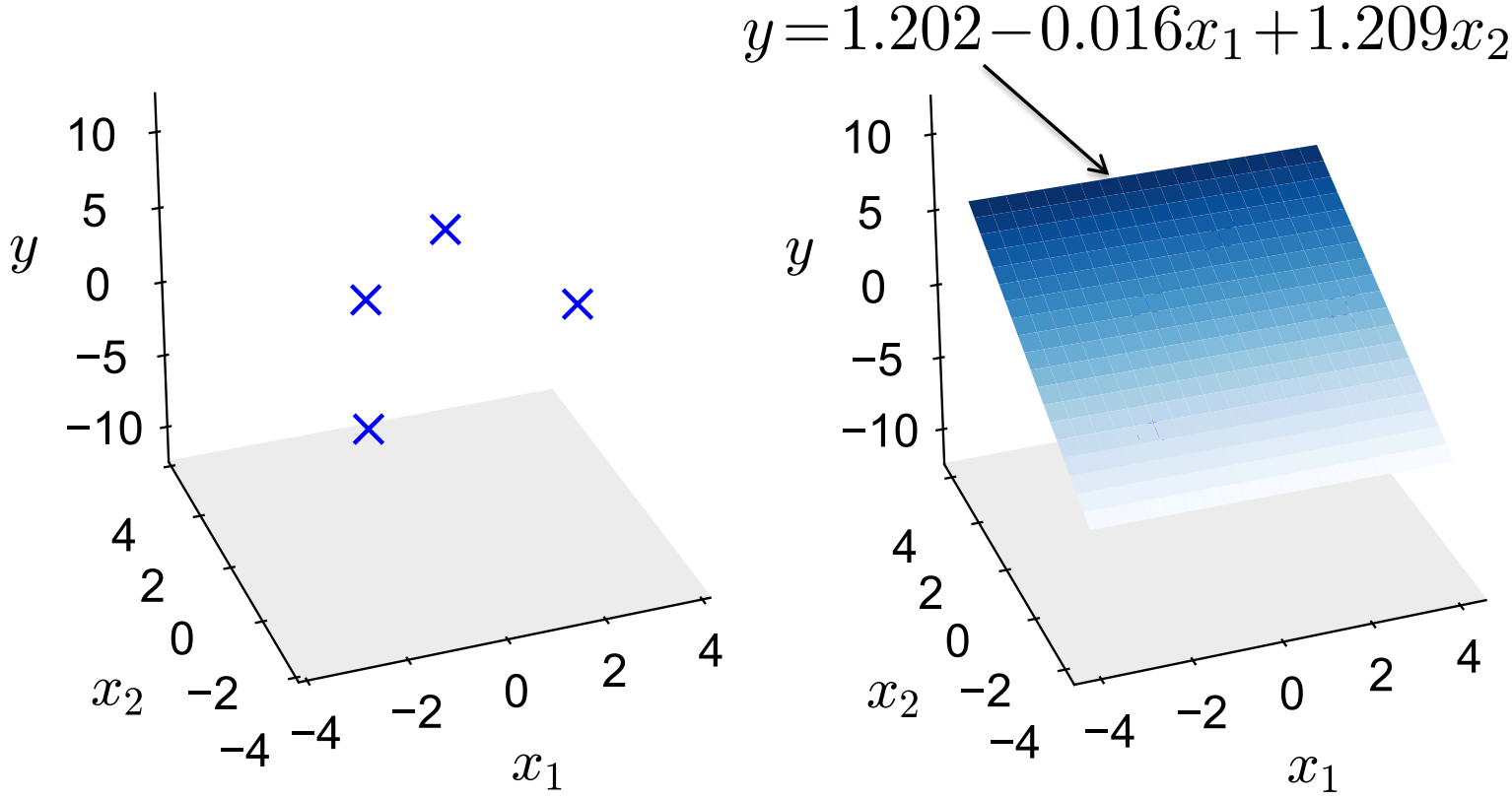
x_1	x_2	y
1	2	4
−1	1	2
3	0	1
−2	−2	−1

$$\mathbf{X} = \begin{pmatrix} 1 & 1 & 2 \\ 1 & -1 & 1 \\ 1 & 3 & 0 \\ 1 & -2 & -2 \end{pmatrix}$$

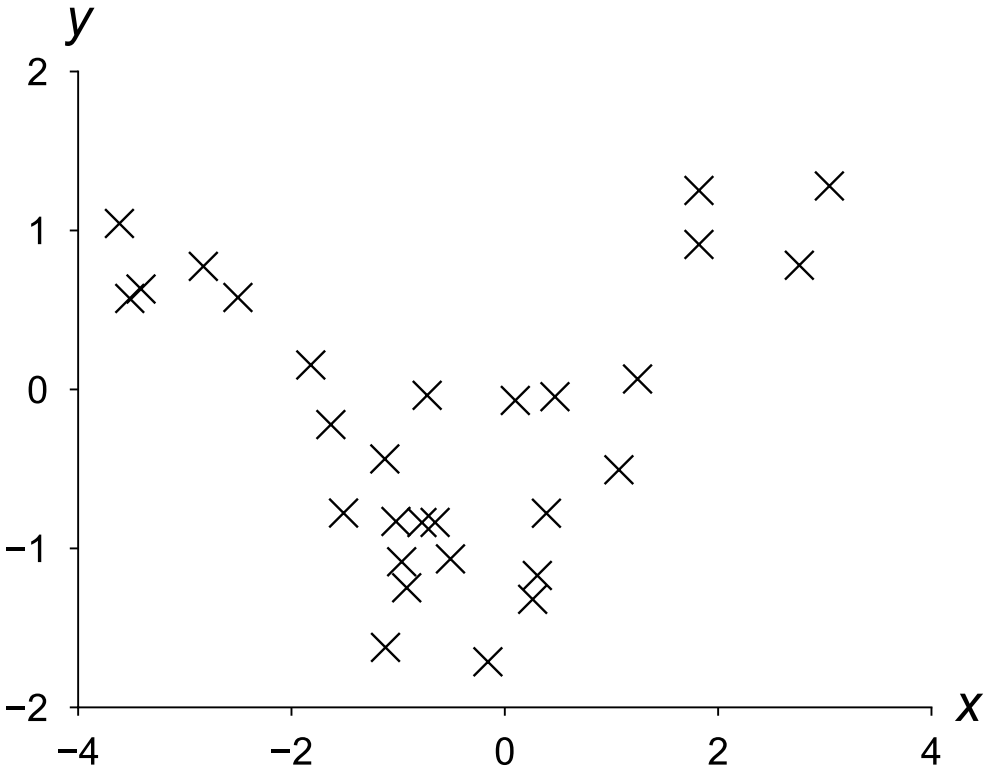
重回帰モデルの計算例 (2)

- よって、重みベクトルwの解は

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = (1.202 \ \ -0.016 \ 1.209)^T$$



もっと複雑にしたい!



- 直線や平面で表せない関係も多いのでは?

関数をもっと複雑にすればよい！

線形回帰モデル

$$y = w_0 + w_1 x + w_2 x^2 = \underbrace{\begin{pmatrix} w_0 & w_1 & w_2 \end{pmatrix}}_{\mathbf{w}^T} \underbrace{\begin{pmatrix} 1 \\ x \\ x^2 \end{pmatrix}}_{\boldsymbol{\phi}(x)}$$

$$y = w_0 + w_1 x + w_2 \sin(x) = \underbrace{\begin{pmatrix} w_0 & w_1 & w_2 \end{pmatrix}}_{\mathbf{w}^T} \underbrace{\begin{pmatrix} 1 \\ x \\ \sin(x) \end{pmatrix}}_{\boldsymbol{\phi}(x)}$$

- どれも、係数ベクトルの線形式として書ける！
  - $y = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$ … 線形回帰モデル (linear regression model)
- これをシグモイド関数に通したのがロジスティック回帰モデル $y = \sigma(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}))$

線形回帰モデル (2)

$$y = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) \ (= \boldsymbol{\phi}(x)^T \mathbf{w})$$

- は、xがφ(x)に変わっただけで重回帰モデル $y = \mathbf{w}^T \mathbf{x}$ と同じなので、たとえば $\boldsymbol{\phi}(x) = (1, x, x^2, x^3)$ のとき、上をN個並べれば

$$\underbrace{\begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_N \end{pmatrix}}_{\hat{\mathbf{y}}} = \begin{pmatrix} \boldsymbol{\phi}(\mathbf{x}_1)^T \\ \boldsymbol{\phi}(\mathbf{x}_2)^T \\ \vdots \\ \boldsymbol{\phi}(\mathbf{x}_N)^T \end{pmatrix} \mathbf{w} = \underbrace{\begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ \vdots & & & \vdots \\ 1 & x_N & x_N^2 & x_N^3 \end{pmatrix}}_{\boldsymbol{\Phi}} \underbrace{\begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{pmatrix}}_{\mathbf{w}}$$

新しい計画行列

線形回帰モデル (3)

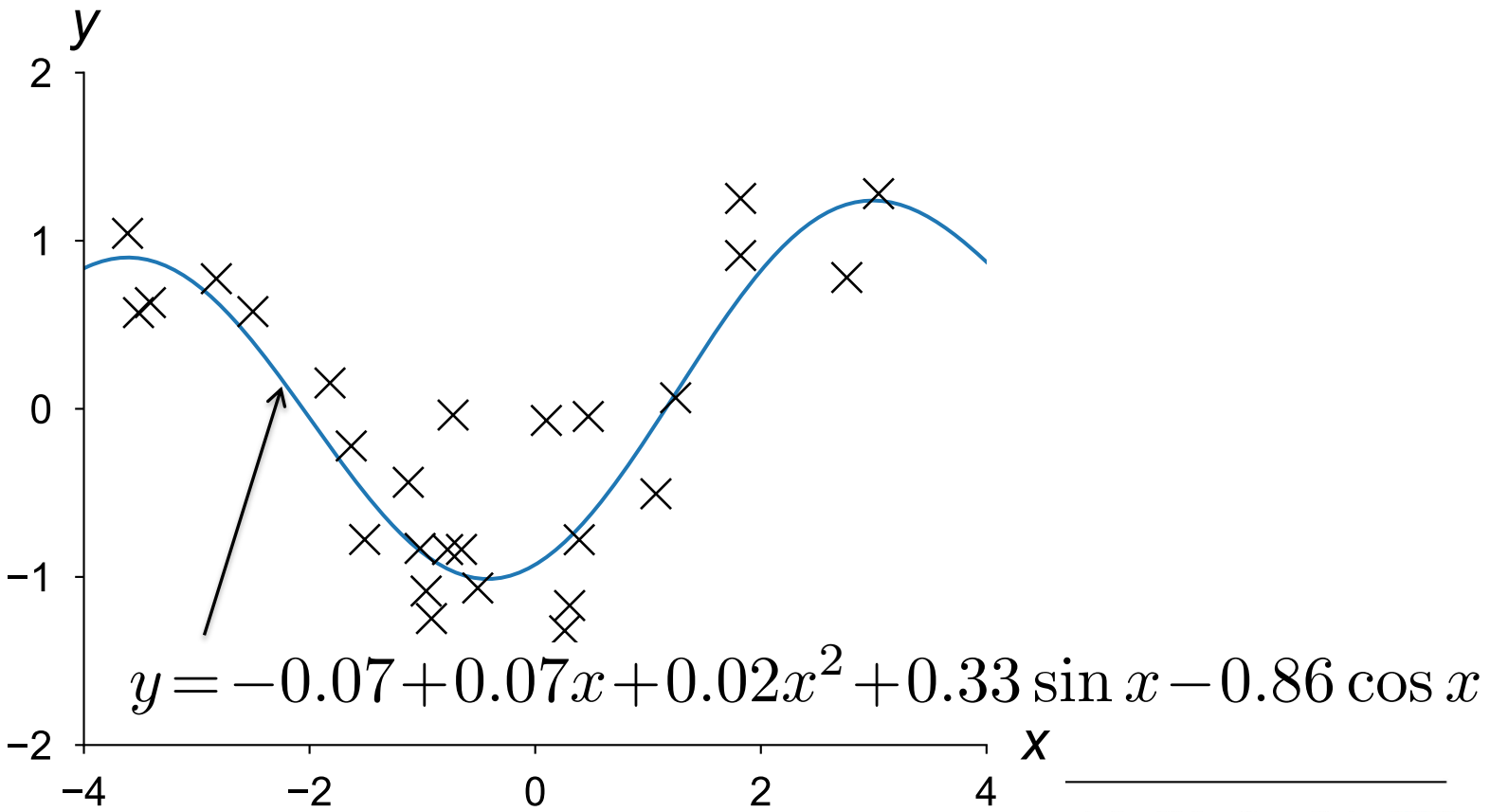
- つまり一般に、線形回帰モデルは以下のように書ける

$$\underbrace{\begin{pmatrix} \widehat{y}_1 \\ \widehat{y}_2 \\ \vdots \\ \widehat{y}_N \end{pmatrix}}_{\widehat{\mathbf{y}}} = \underbrace{\begin{pmatrix} 1 & \phi_1(x_1) & \cdots & \phi_H(x_1) \\ 1 & \phi_1(x_2) & \cdots & \phi_H(x_2) \\ \vdots & & & \vdots \\ 1 & \phi_1(x_N) & \cdots & \phi_H(x_N) \end{pmatrix}}_{\mathbf{\Phi}} \underbrace{\begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_H \end{pmatrix}}_{\mathbf{w}}$$

- 計画行列Φを使って、$\widehat{\mathbf{y}} = \mathbf{\Phi}\mathbf{w}$ と書ける
- $\mathbf{X} \mapsto \mathbf{\Phi}$ 以外は重回帰と同じなので、$\mathbf{w} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{y}$

線形回帰モデルの例

- 特徴ベクトルを $\boldsymbol{\phi}(x) = (1, x, x^2, \sin x, \cos x)^T$ として先ほどのデータに適用すると、
  $\mathbf{w} = (-0.065, 0.068, 0.022, 0.333, -0.863)^T$ が解



線形回帰モデルと基底関数

$$y = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$$

$$\mathbf{w} = (w_0, w_1, w_2, \cdots, w_H)$$

$$\boldsymbol{\phi}(\mathbf{x}) = (\underbrace{\phi_0(\mathbf{x})}_{=1}, \phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \cdots, \phi_H(\mathbf{x}))$$

- よって、$y = w_0 + w_1\phi_1(\mathbf{x}) + w_2\phi_2(\mathbf{x}) + \cdots + w_H\phi_H(\mathbf{x})$ は関数　$y = \phi_0(\mathbf{x})\ (= 1)$

  $y = \phi_1(\mathbf{x})$

  $y = \phi_2(\mathbf{x})$

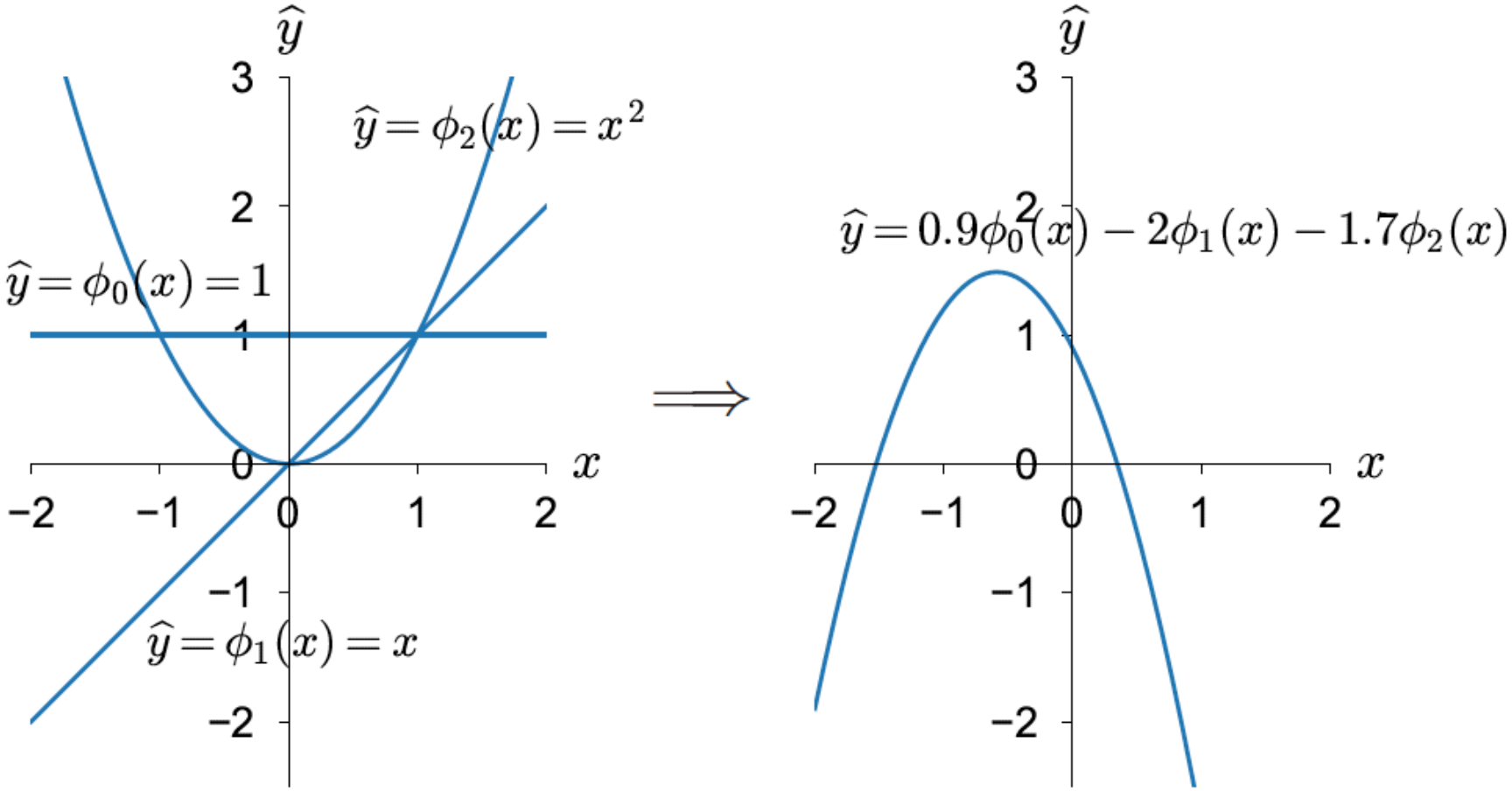
  $\vdots$

  $y = \phi_H(\mathbf{x})$

  の重みつき和 (線形結合) とみなせる

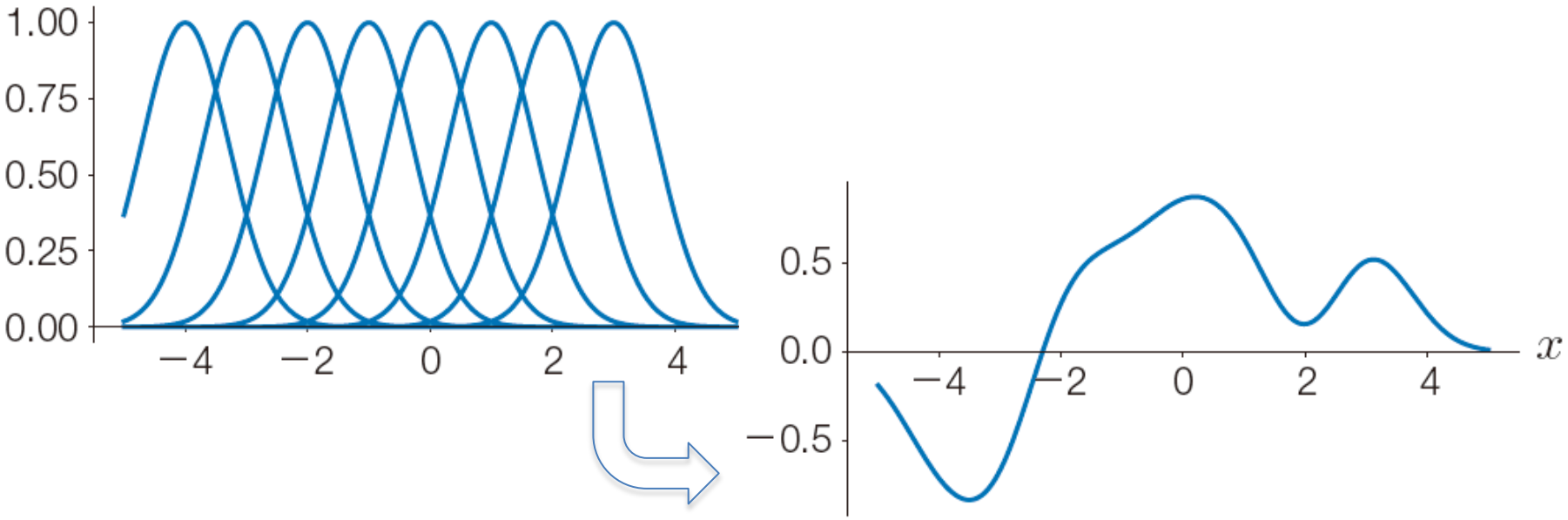
基底関数という

線形回帰モデルと基底関数



- 2次関数$y=-1.7x^2-2x+0.9$ は、関数 $y=1$, $y=x$, $y=x^2$の線形和

動径基底関数回帰

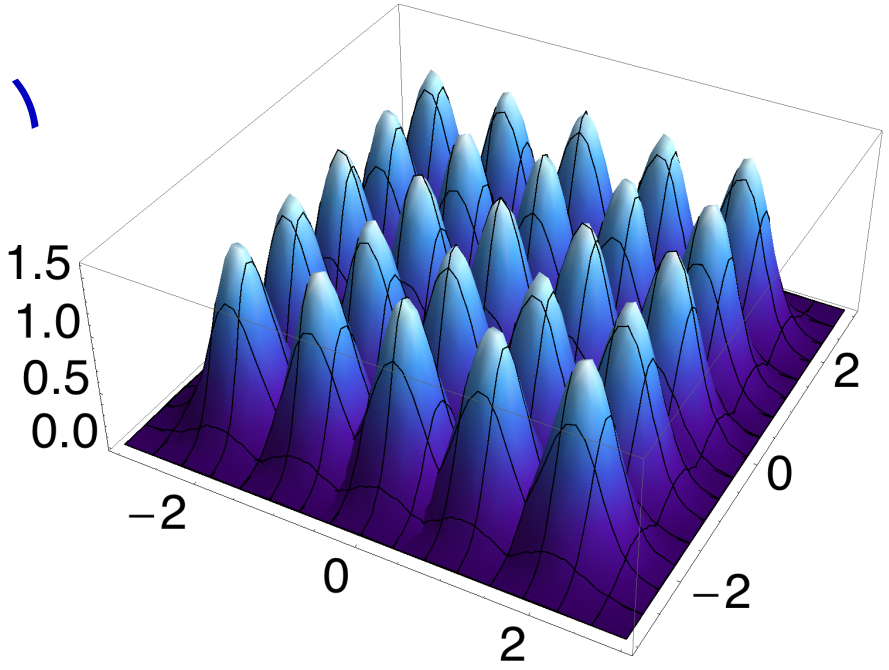


- それなら、$\phi_h(x) = \exp\left(-\frac{(x-\mu_h)^2}{\sigma^2}\right)$ をたくさん用意すれば、任意の関数が表せるのでは?

動径基底関数回帰 (radial basis function regression)

次元の呪い

- しかし…



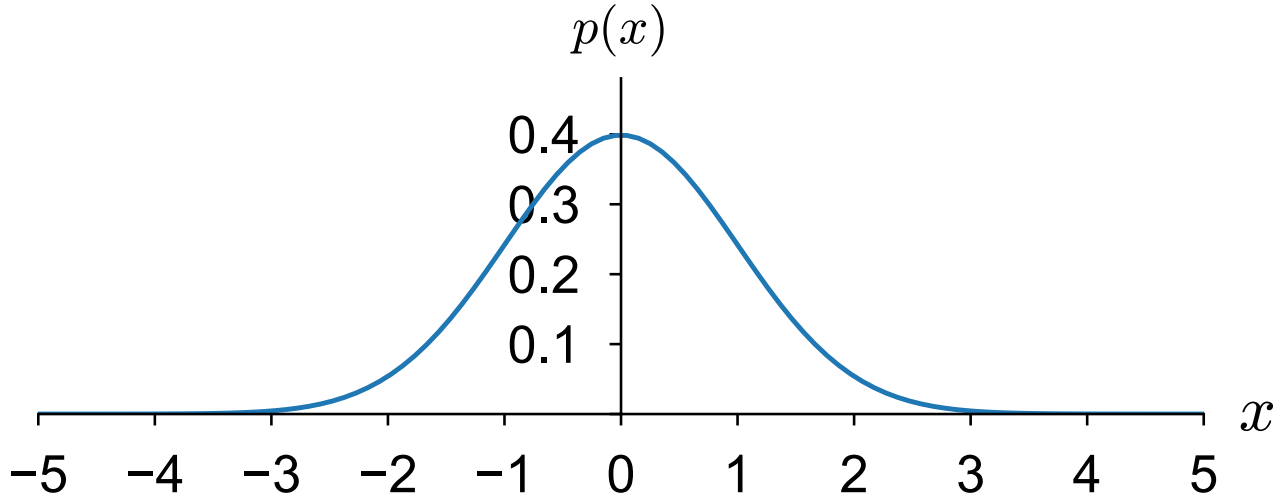
- 動径基底関数回帰に必要な基底関数の数(=パラメータの数)は、xの次元が増えると指数的に増加
  - 1おきに基底関数をとると、[-10,10]で1次元では21個
  - 2次元では21^2=441個
  - 10次元では21^{10}=16,679,880,978,201個！

  → 次元の呪い (curse of dimensionality)

ガウス分布とガウス過程

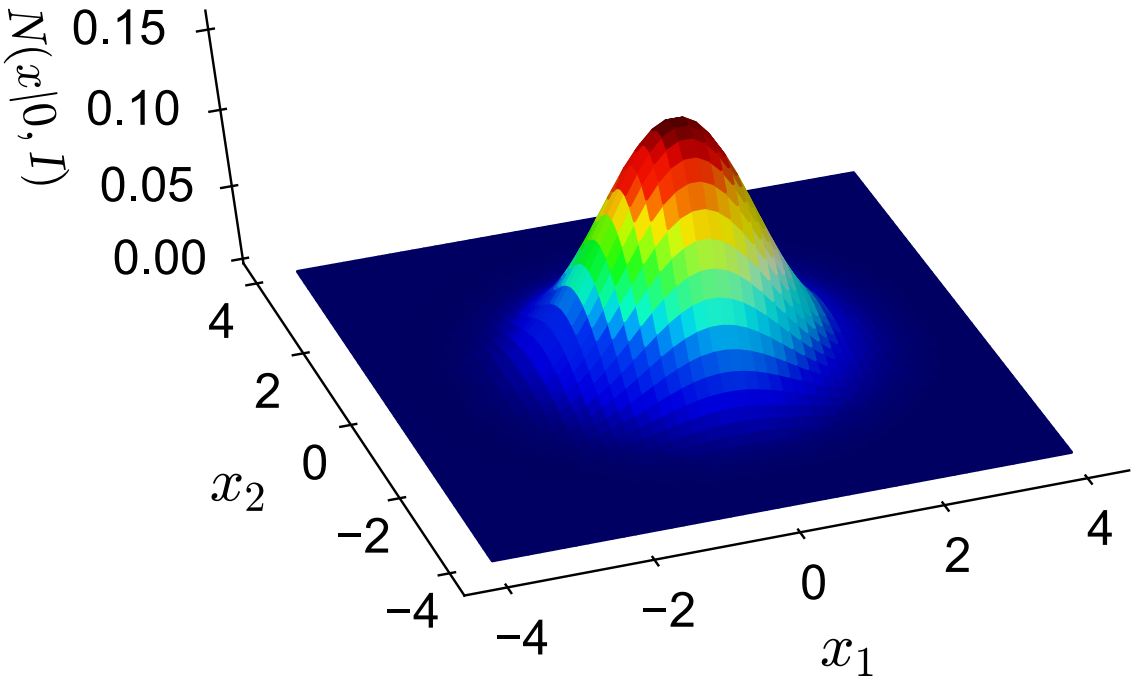
ガウス分布

- ガウス分布 (Gaussian distribution) または
  正規分布 (normal distribution)：最も基本的な確率分布



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

多変量ガウス分布

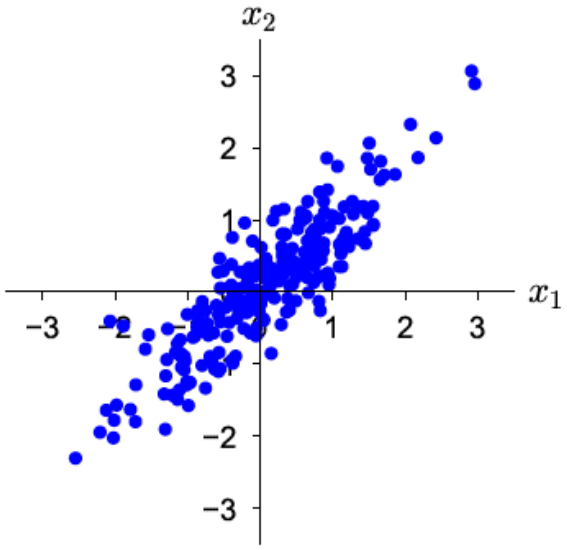


- $p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \dfrac{1}{(\sqrt{2\pi})^D \sqrt{|\boldsymbol{\Sigma}|}} \exp\left(-\dfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$

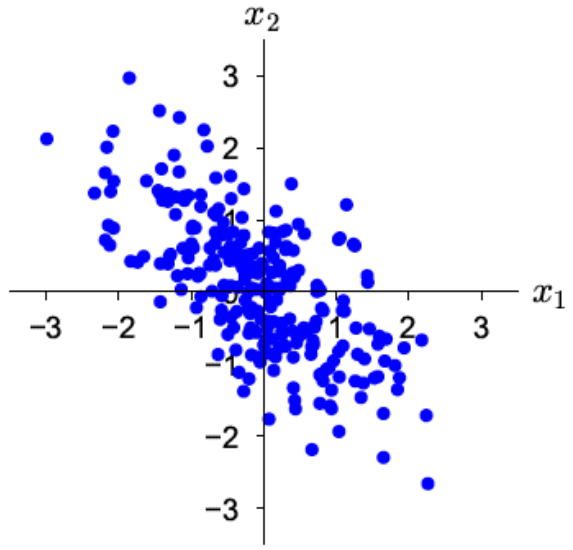
$$\boldsymbol{\mu} = \mathbb{E}[\mathbf{x}] \qquad \text{(平均ベクトル)}$$

$$\boldsymbol{\Sigma} = \mathbb{E}[\mathbf{x}\mathbf{x}^T] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{x}]^T \qquad \text{(共分散行列)}$$

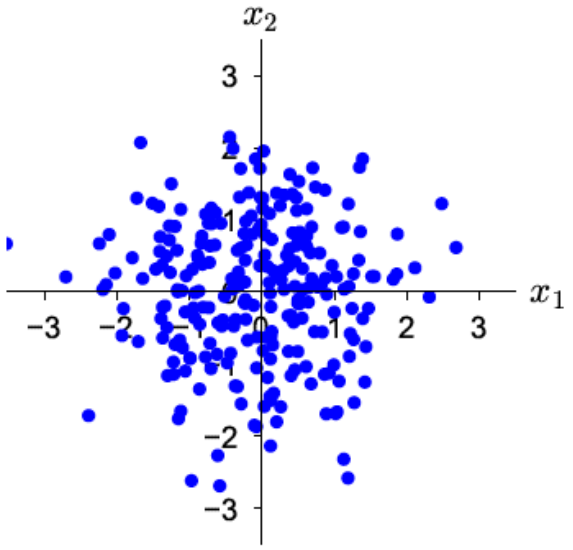
多変量ガウス分布からのサンプル



(a) $\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$



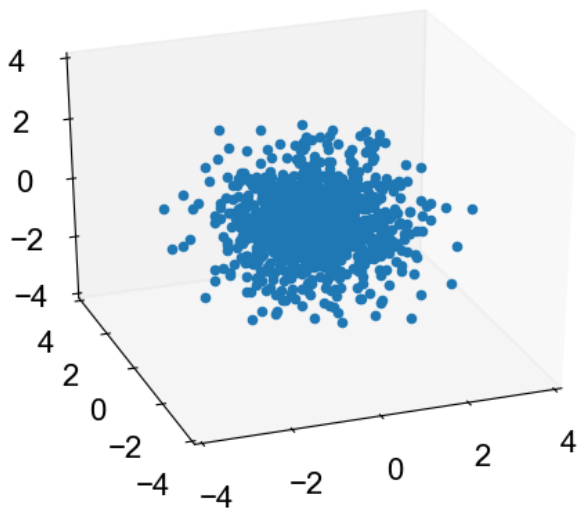
(b) $\boldsymbol{\Sigma} = \begin{pmatrix} 1 & -0.7 \\ -0.7 & 1 \end{pmatrix}$

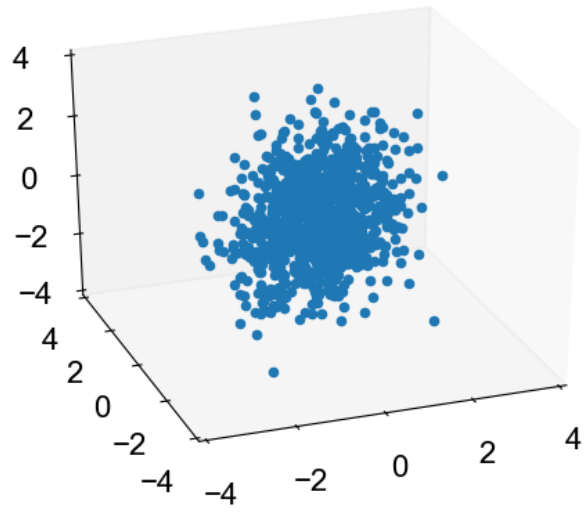


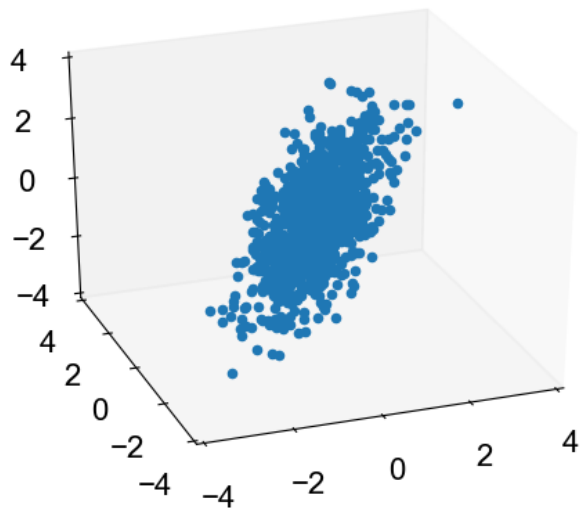
(c) $\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

- 共分散行列の要素の値が大きい(共分散が大)と、類似した値がサンプルされる
  - 負では逆相関、0ならば、無相関

多変量ガウス分布からのサンプル (2)







$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.5 & 0.2 \\ 0.5 & 1 & 0.5 \\ 0.2 & 0.5 & 1 \end{pmatrix} \quad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.8 & 0.6 \\ 0.8 & 1 & 0.8 \\ 0.6 & 0.8 & 1 \end{pmatrix}$$

- 3次元の場合

多変量ガウス分布の線形変換

- $\mathbf{x}$ が多変量ガウス分布に従っていて

$$p(\mathbf{x}) \propto \exp\left(-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}\right)$$

のとき、

$\mathbf{x}=\mathbf{A}^{-1}\mathbf{y}$

- $\mathbf{x}$ を行列 $\mathbf{A}$で変換した $\mathbf{y}=\mathbf{A}\mathbf{x}$ の分布は

$$p(\mathbf{y}) \propto \exp\left(-\frac{1}{2}(\mathbf{A}^{-1}\mathbf{y})^T\boldsymbol{\Sigma}^{-1}(\mathbf{A}^{-1}\mathbf{y})\right)\left|\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right|$$

$$\propto \exp\left(-\frac{1}{2}\mathbf{y}^T\boldsymbol{\Lambda}\mathbf{y}\right) \quad \left(\boldsymbol{\Lambda}=(\mathbf{A}^{-1})^T\boldsymbol{\Sigma}^{-1}\mathbf{A}^{-1}\mathbf{y}\right)$$

  - よって、$\mathbf{y}$ もガウス分布に従う

線形回帰モデルふたたび

- 線形回帰モデル $\mathbf{y} = \mathbf{\Phi w}$ において、重みベクトル w がガウス分布

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \alpha \mathbf{I})$$

に従っているとする

$$\underbrace{\begin{pmatrix} \widehat{y}_1 \\ \widehat{y}_2 \\ \vdots \\ \widehat{y}_N \end{pmatrix}}_{\widehat{\mathbf{y}}} = \underbrace{\begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_H(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_H(\mathbf{x}_2) \\ \vdots & & & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_H(\mathbf{x}_N) \end{pmatrix}}_{\mathbf{\Phi}} \underbrace{\begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_H \end{pmatrix}}_{\mathbf{w}}$$

重みwの積分消去

- このとき、Φは定数行列なので、wを定数行列で変換した $\mathbf{y}=\mathbf{\Phi w}$ もガウス分布に従い、
  - 平均　　$\boldsymbol{\mu}=\mathbb{E}[\mathbf{y}]=\mathbb{E}[\mathbf{\Phi w}]=\mathbf{\Phi}\mathbb{E}[\mathbf{w}]=\mathbf{0}$
  - 共分散　$\mathbf{\Sigma}=\mathbb{E}[\mathbf{y}\mathbf{y}^T]-\mathbb{E}[\mathbf{y}]\mathbb{E}[\mathbf{y}]^T$
    $=\mathbb{E}[(\mathbf{\Phi w})(\mathbf{\Phi w})^T]=\mathbf{\Phi}\mathbb{E}[\mathbf{w}\mathbf{w}^T]\mathbf{\Phi}^T$
    $=\alpha\mathbf{\Phi}\mathbf{\Phi}^T$

- すなわち、yは全体として、

$$\mathbf{y}\sim\mathcal{N}(\mathbf{0},\alpha\mathbf{\Phi}\mathbf{\Phi}^T)$$

のガウス分布に従う.

重みwの積分消去 (2)

$$\underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}}_{\mathbf{y}} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \alpha \underbrace{\begin{pmatrix} \phi_0(\mathbf{x}_1) \cdots \phi_H(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) \cdots \phi_H(\mathbf{x}_2) \\ \vdots \\ \phi_0(\mathbf{x}_N) \cdots \phi_H(\mathbf{x}_N) \end{pmatrix}}_{\mathbf{\Phi}} \underbrace{\begin{pmatrix} \phi_0(\mathbf{x}_1) \cdots \phi_0(\mathbf{x}_N) \\ \phi_1(\mathbf{x}_1) \cdots \phi_1(\mathbf{x}_N) \\ \vdots \\ \phi_H(\mathbf{x}_1) \cdots \phi_H(\mathbf{x}_N) \end{pmatrix}}_{\mathbf{\Phi}^T} \right)$$

- 線形回帰モデル

$$\mathbf{y} = \mathbf{\Phi}\mathbf{w}$$

が、ガウス分布に従う重みwを積分消去して

$$\mathbf{y} \sim \mathcal{N}\left(\mathbf{0}, \alpha \mathbf{\Phi}\mathbf{\Phi}^T\right)$$

になった

ガウス過程

$$p(\mathbf{y}|\mathbf{X}) = \mathcal{N}(\mathbf{0}, \alpha\mathbf{\Phi}\mathbf{\Phi}^T)$$

は、どんな入力 $\mathbf{X}=\{\mathbf{x}_n\}_{n=1}^N$ についても成り立つ
→ ガウス過程

- どんな入力 $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N)$ についても、対応する出力 $\mathbf{y}=(y_1, y_2, \cdots, y_N)$ がガウス分布に従うとき、$\mathbf{y}$ はガウス過程に従う、という
  - ガウス過程＝無限次元のガウス分布
  - 線形回帰モデルで、重みwを積分消去したもの
- $\mathbf{K} = \alpha\mathbf{\Phi}\mathbf{\Phi}^T$ の要素を与えるカーネル関数

  $$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \alpha\boldsymbol{\phi}(\mathbf{x}_i)^T\boldsymbol{\phi}(\mathbf{x}_j)$$

  だけでガウス分布が定まる (カーネル法, SVMと同じ)

ガウス過程 (2)

$$\mathbf{K} = \lambda^2 \mathbf{\Phi}\mathbf{\Phi}^T = \lambda^2 \underbrace{\begin{pmatrix} \vdots \\ \boxed{\phi(\mathbf{x}_n)^T} \\ \vdots \end{pmatrix}}_{\mathbf{\Phi}} \underbrace{\begin{pmatrix} \cdots & \boxed{\phi(\mathbf{x}_{n'})} & \cdots \end{pmatrix}}_{\mathbf{\Phi}^T}$$

- x_nと$x_{n'}$が近ければ、共分散行列Kの要素$K_{nn'}$も大きい
  ↓
  y_n, $y_{n'}$が近い値をとる
- ガウス過程は、xが似ていればyも似ている ことを数学的に表すための確率過程.

カーネルと特徴ベクトル

- ガウス過程では、$K_{ij} = \alpha \boldsymbol{\phi}(\mathbf{x}_i)^T \boldsymbol{\phi}(\mathbf{x}_j)$ だけが必要

  ↓

  $\boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)$ を直接求める必要はない

  – 例： $k(\mathbf{x}, \mathbf{x}') = (x_1 x_1' + x_2 x_2' + 1)^2$ のとき
  $\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$ となるが、
  この $\phi(\mathbf{x})$ を計算する必要はない

- $\phi(\mathbf{x})$ を求めると、無限次元になることもある
  (＝無限次元の線形回帰モデルに相当)
- これをカーネルトリックという

さまざまなカーネル

- 線形カーネル: $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$
  - $\phi(\mathbf{x}) = \mathbf{x}$ を意味する→ガウス過程は、重回帰を包む
- ガウスカーネル:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{|\mathbf{x}-\mathbf{x}'|^2}{\theta}\right)$$

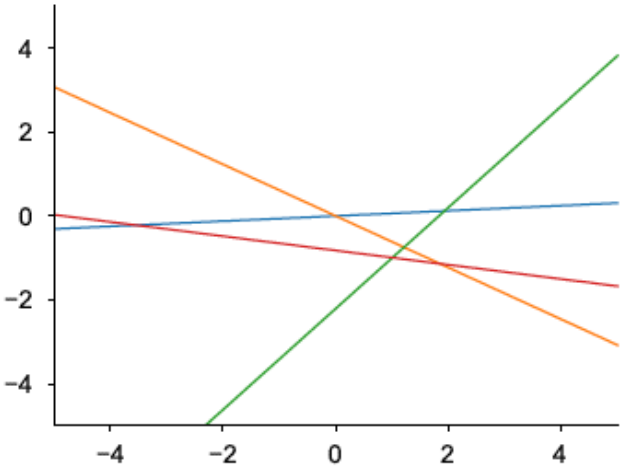
- 指数カーネル:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{|\mathbf{x}-\mathbf{x}'|}{\theta}\right)$$

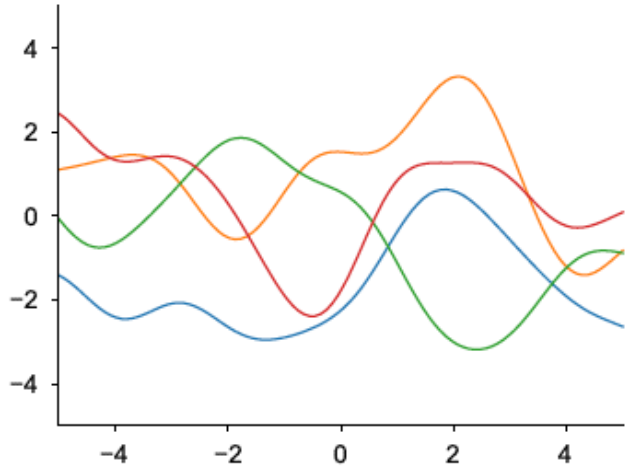
- 周期カーネル:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(\cos\theta_1 \left(\frac{|\mathbf{x}-\mathbf{x}'|}{\theta_2}\right)\right)$$

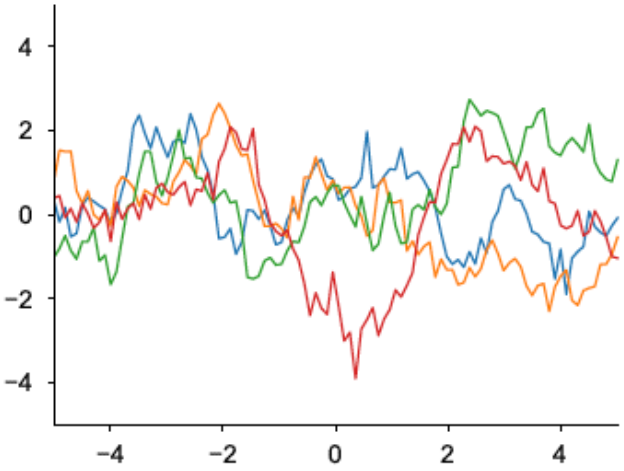
さまざまなカーネル (2)



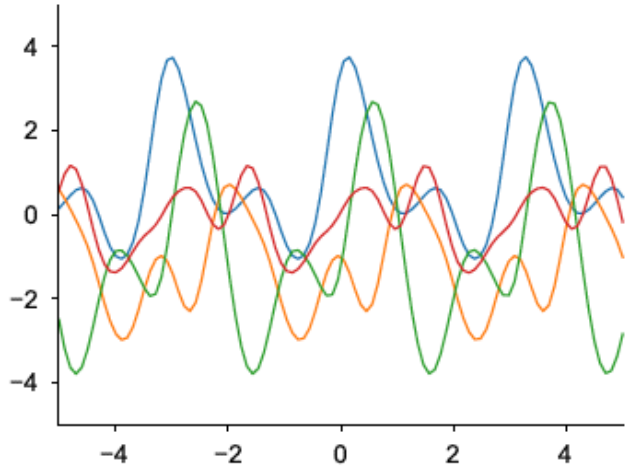
(a) 線形カーネル: $\mathbf{x}^T\mathbf{x}'$



(b) ガウスカーネル: $\exp(-|\mathbf{x}-\mathbf{x}'|^2/\theta)$



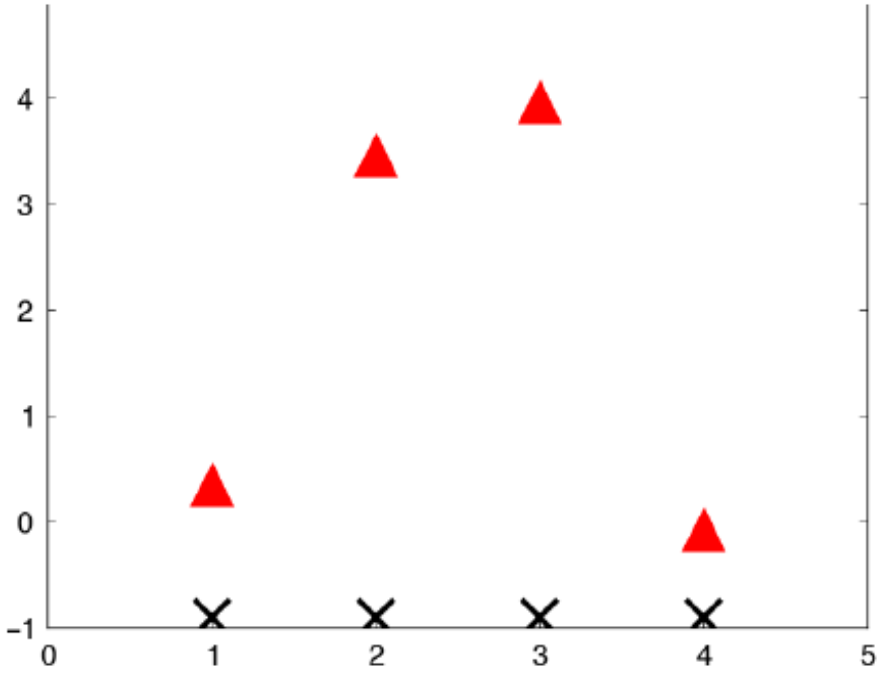
(c) 指数カーネル: $\exp(-|\mathbf{x}-\mathbf{x}'|/\theta)$
(Ornstein-Uhlenbeck 過程)



(d) 周期カーネル: $\exp(\theta_1 \cos(|\mathbf{x}-\mathbf{x}'|/\theta_2))$

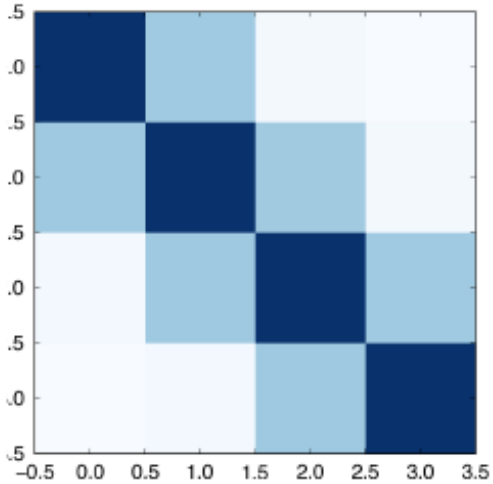
直感的理解

- 相関のある多変量ガウス分布



ガウス分布からのサンプル

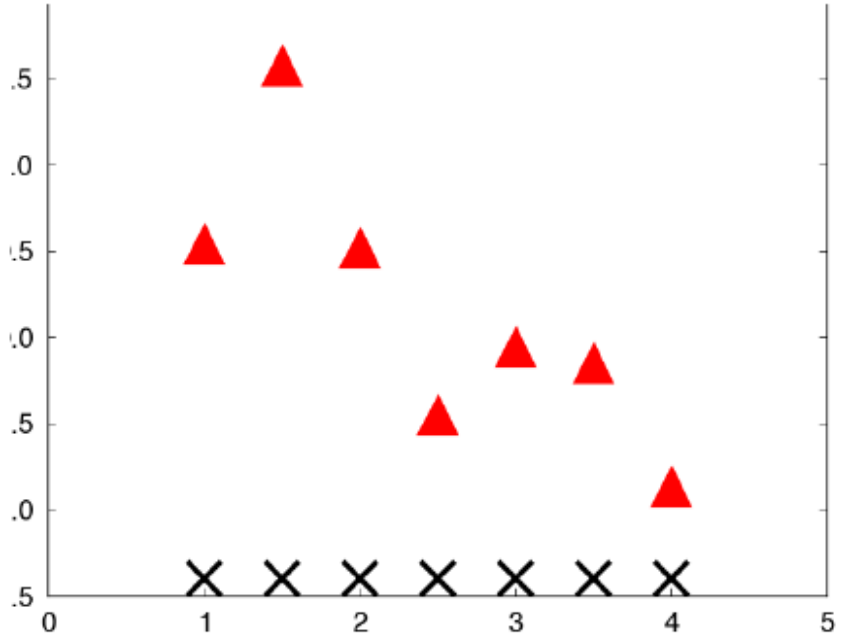
$K =$



分散・共分散行列

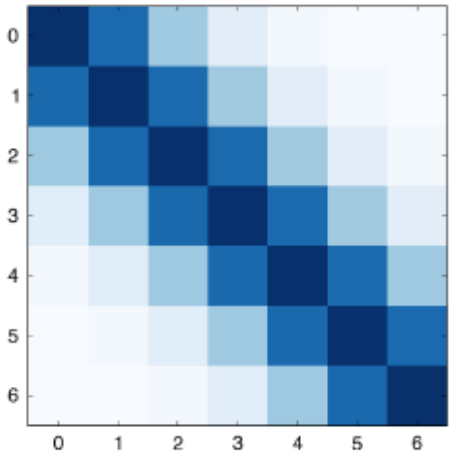
直感的理解

- 相関のある多変量ガウス分布



ガウス分布からのサンプル

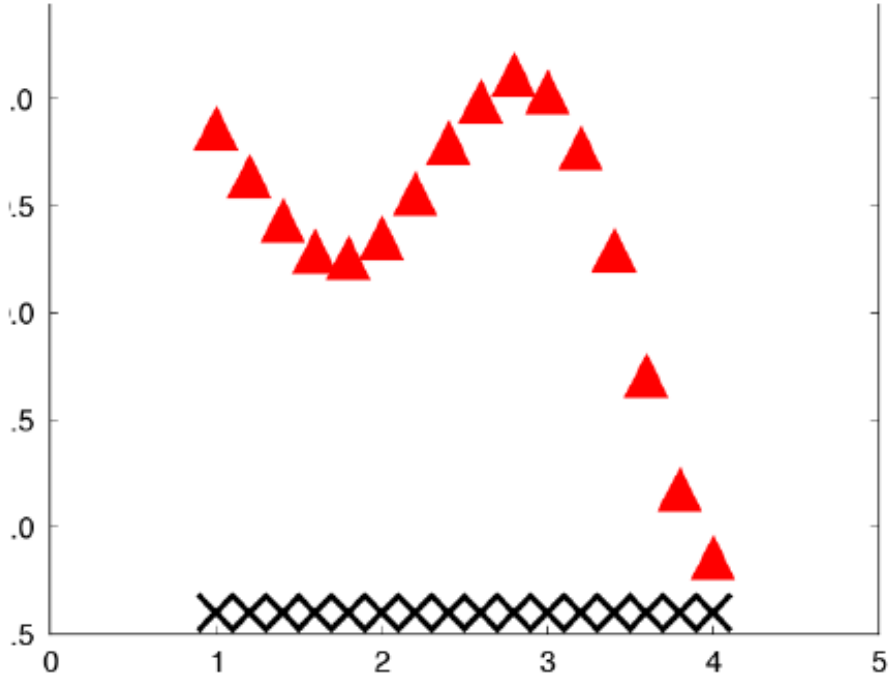
$K =$



分散・共分散行列

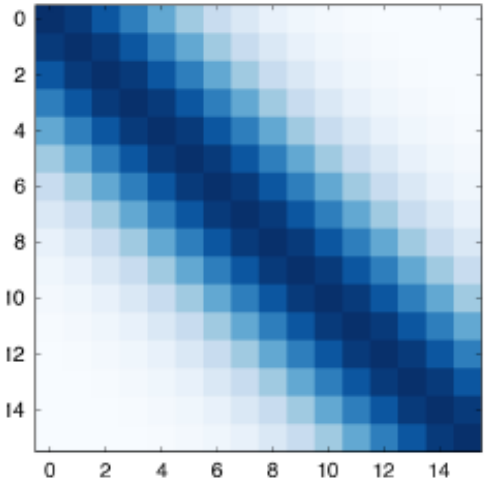
直感的理解

- 相関のある多変量ガウス分布



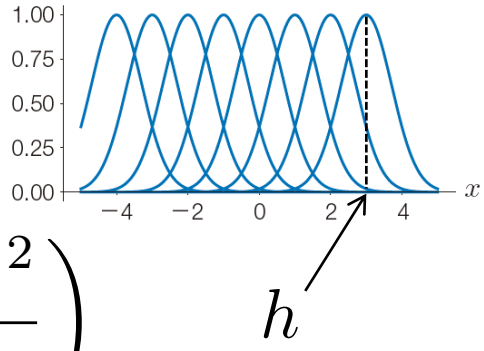
ガウス分布からのサンプル

$K =$



分散・共分散行列

「基底関数」の消去



- 点hでの基底関数

$$\phi_h(x) = \tau \exp\left(-\frac{(x - h/H)^2}{r^2}\right)$$

を考えてみる

- $H \to \infty$にしてグリッドを無限に細かくすると、

$$k(x, x') = \lim_{H\to\infty} \sum_{h=-H^2}^{H^2} \phi_h(x)\phi_h(x')$$

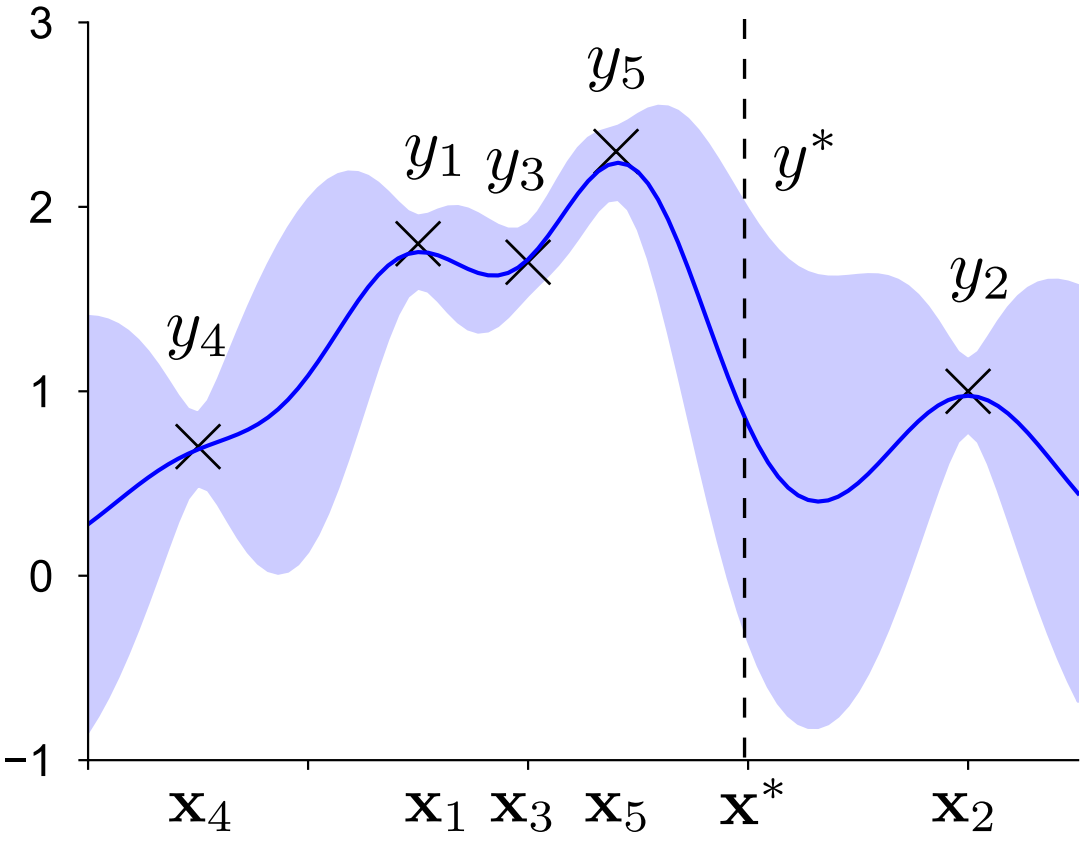
$$\to \int_{-\infty}^{\infty} \tau^2 \exp\left(-\frac{(x-h)^2}{r^2}\right) \exp\left(-\frac{(x'-h)^2}{r^2}\right) dh$$

$$= \tau^2 \sqrt{\pi r^2/2}\ \exp\left(-\frac{1}{2r^2}(x-x')^2\right)$$

$$\equiv \boxed{\theta_1 \exp\left(-\frac{1}{\theta_2}(x-x')^2\right)}$$ ガウスカーネル！

ガウス過程回帰モデル

- 新しい入力点x*での出力y*の分布はどうなるか？



ガウス過程回帰モデル (2)

- 学習データのyにy*を加えた y'=(y, y*) が、
  学習データのXにx*を加えたX'から計算される行列
  を共分散行列としたガウス分布に従うので

$$\begin{pmatrix} y_1 \\ \vdots \\ y_N \\ y^* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \begin{array}{c} \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \\ \mathbf{x}^* \end{array} \begin{array}{c} \begin{array}{ccc} \mathbf{x}_1 & \cdots & \mathbf{x}_N \end{array} \quad \mathbf{x}^* \\ \begin{pmatrix} \mathbf{K} & \mathbf{k}_* \\ \mathbf{k}_*^T & k_{**} \end{pmatrix} \end{array} \right)$$

ここで $\mathbf{k}_* = (k(x^*, x_1), k(x^*, x_2), \cdots, k(x^*, x_N))$

$k_{**} = k(x^*, x^*)$

ガウス過程回帰モデル (3)

- 数式で簡潔に書くと

$$\begin{pmatrix} \mathbf{y} \\ y^* \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} \mathbf{K} & \mathbf{k}_* \\ \mathbf{k}_*^T & k_{**} \end{pmatrix}\right)$$

- なので、多変量ガウス分布の条件つき分布の公式から

$$p(y^*|\mathbf{x}^*, \mathbf{X}, \mathbf{y}) = \mathcal{N}\left(\mathbf{k}_*^T\mathbf{K}^{-1}\mathbf{y},\ k_{**} - \mathbf{k}_*^T\mathbf{K}^{-1}\mathbf{k}_*\right)$$

  - よって、その期待値は

$$\mathbb{E}[y^*|\mathbf{x}^*, \mathbf{X}, \mathbf{y}] = \mathbf{k}_*^T\mathbf{K}^{-1}\mathbf{y}$$

条件付きガウス分布の公式

- 多変量ガウス分布の条件付き分布は、

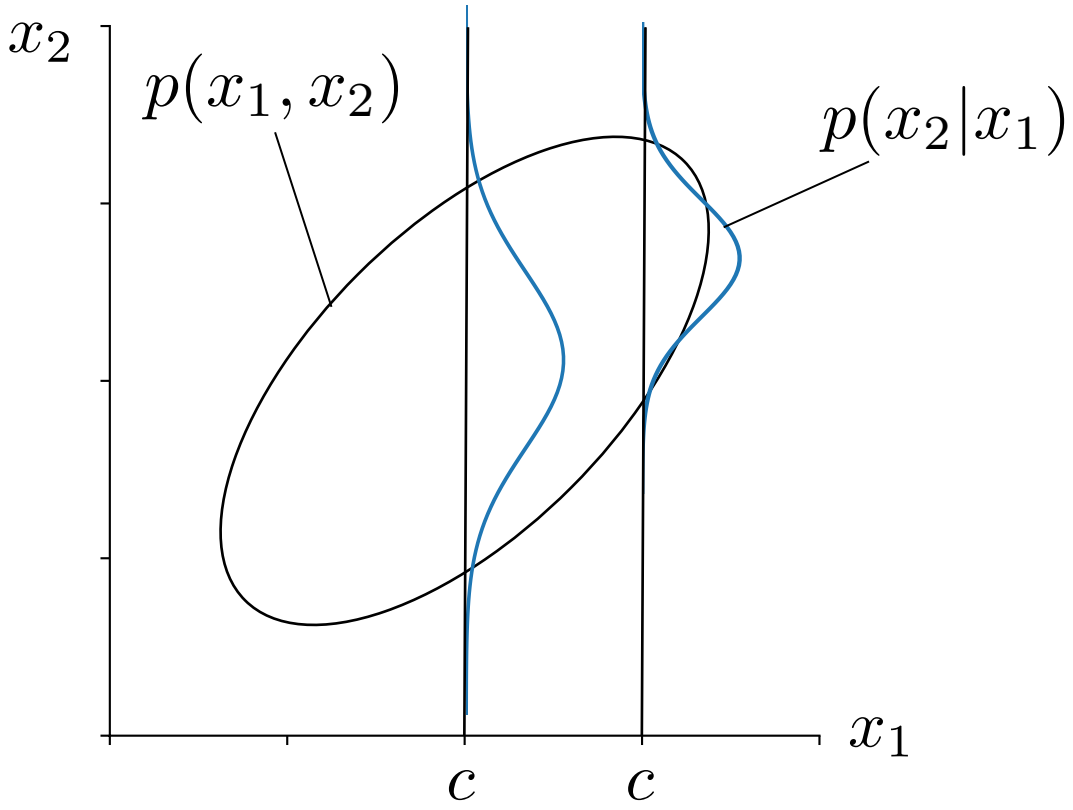
$$\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix} \right)$$

  のとき

$$p(\mathbf{x}_2|\mathbf{x}_1) = \mathcal{N}\left(\boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}(\mathbf{x}_1 - \boldsymbol{\mu}_1), \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}\right)$$

- 証明は教科書を参照してください

条件付きガウス分布のイメージ



- 多変量ガウス分布 $p(x_1, x_2)$ を x_1 で条件づけると、「切り口」$p(x_2|x_1)$ はまたガウス分布になる

ガウス過程回帰モデル (4)

- 注：ガウス過程回帰の期待値

$$\mathbb{E}[y^*|\mathbf{x}^*, \mathbf{X}, \mathbf{y}] = \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{y}$$

はカーネルリッジ回帰と実は同じだが、カーネル法と異なりベイズ推定なので、

- 分散も使って分布を推定することができ、

$$p(y^*|\mathbf{x}^*, \mathbf{X}, \mathbf{y}) = \mathcal{N}\left(\mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{y},\, k_{**} - \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{k}_*\right)$$

- カーネル法と違って完全な確率モデルなので、カーネル自体を学習することも可能
  (通常のカーネル法ではクロスバリデーションに頼る必要があり、最適化は難しい)

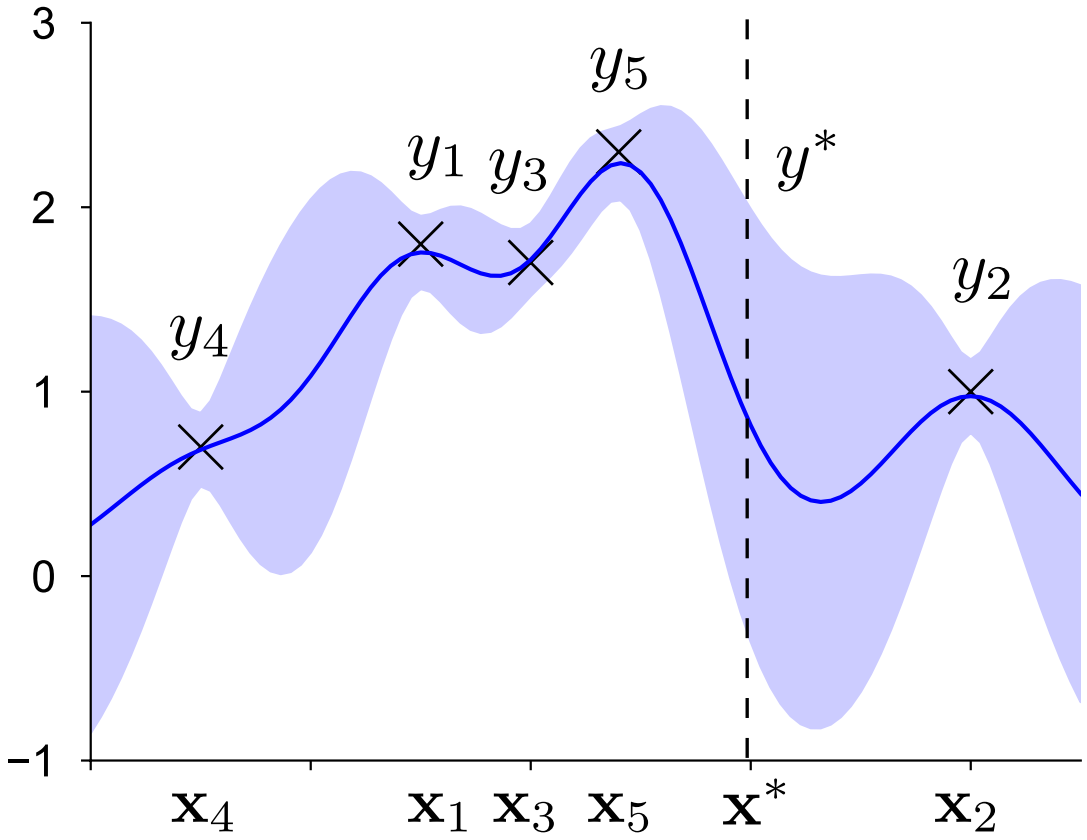
ガウス過程回帰のアルゴリズム

1: [mu,var] = gpr (xtest, xtrain, ytrain, kernel)
2: N = length (ytrain)
3: **for** $n = 1 \cdots N$ **do**
4: **for** $n' = 1 \cdots N$ **do**
5: $K[n, n'] = \text{kernel}\,(\text{xtrain}[n], \text{xtrain}[n'])$
6: **end for**
7: **end for**
8: yy = $K^{-1} *$ ytrain
9: **for** $m = 1 \cdots M$ **do**
10: **for** $n = 1 \cdots N$ **do**
11: $k[n] = \text{kernel}\,(\text{xtrain}[n], \text{xtest}[m])$
12: **end for**
13: $s = \text{kernel}\,(\text{xtest}[m], \text{xtest}[m])$
14: mu$[m] = k *$ yy
15: var$[m] = s - k * K^{-1} * k^T$
16: **end for**

入力:
xtrain $= [\mathbf{x}_1, \ldots, \mathbf{x}_N]$
– 入力 $\mathbf{x} \in \mathbb{R}^D$ を N 個並べたベクトル.
ytrain $= [y_1, \ldots, y_N]^T$
– 出力 $y \in \mathbb{R}$ を N 個並べたベクトル.
xtest $= [\mathbf{x}'_1, \ldots, \mathbf{x}'_M]$
– 回帰したい入力 $\mathbf{x}'$ を M 個並べたベクトル.
出力:
mu : xtest に対応する y の期待値
var : xtest に対応する y の分散

- (xtrain,ytrain)が与えられたとき、xtestの各点について平均muと分散varを出力

ガウス過程回帰の例



- 新しい入力x*での予測値y*の分布は、ガウス分布
- 青線は期待値、水色の領域は±2σのエリア

ガウス過程回帰と自然言語処理

- 機械翻訳の品質推定 (Cohn+ 2013)
- 作文の自動採点
- 文の処理時間のモデル化
  - 特に、分散を捉えることが重要！
  - 線形回帰やSVRでは無理

これらも
同じ問題

Modelling Annotator Bias with Multi-task Gaussian Processes: An Application to Machine Translation Quality Estimation

Trevor Cohn and **Lucia Specia**
Department of Computer Science
University of Sheffield
Sheffield, United Kingdom
{t.cohn,l.specia}@sheffield.ac.uk

機械翻訳の評価 (Cohn+2013)

- 文を与えて、その品質(例えばTOEICスコア)を出力する問題
  “*Would you mind helping me?*” → 652
  - 「文の処理時間」を出力する問題とも同じ
- 回帰モデルは一般に非線形
  - 少しの間違いで、大幅にスコア低下のことも
  - 短い文では、評価は本来曖昧…分散が重要
  - ニューラルでは、突然変な値を出す可能性がある
  - 類似する文は類似するスコア→「類似」を定義できればよい
    → ガウス過程そのもの！

機械翻訳の評価 (2) (Cohn+2013)

- 文スコアの推定を、ガウス過程回帰としてとらえる

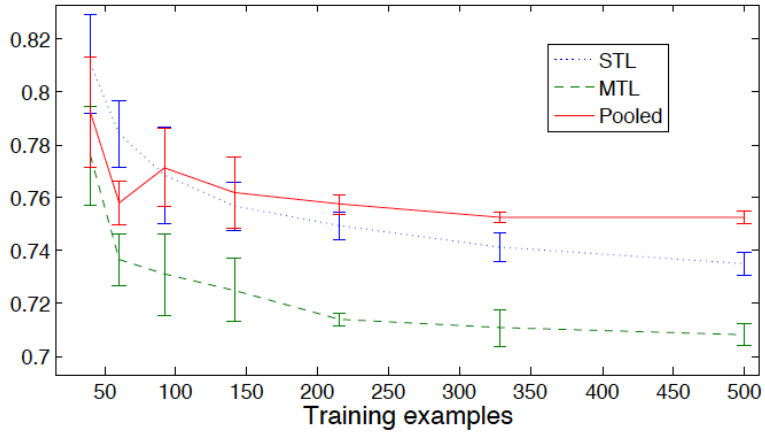
$$y_* \sim \mathcal{N}\big(\mathbf{k}_*^T(K+\sigma_n^2 I)^{-1}\mathbf{y}, \\ k(\mathbf{x}_*,\mathbf{x}_*) - \mathbf{k}_*^T(K+\sigma_n^2 I)^{-1}\mathbf{k}_*\big)$$

- 注: ニューラル手法 (例えばTakahashi+ ACL2020)を取り込むことも可能
  - ガウス過程の入力特徴をニューラルで学習 (Neural CRFと同様のアイデア)
  - 出力はガウス過程なので、解析的(解釈可能)かつ分散が求められる
  - Neural Network GP (Wilson+2016他多数)

機械翻訳の評価 (2) (Cohn+2013)

- Support Vector Regressionより高性能

Model	MAE	RMSE
μ	0.8279	0.9899
SVM	0.6889	0.8201
Linear ARD	0.7063	0.8480
Squared exp. Isotropic	0.6813	0.8146
Squared exp. ARD	**0.6680**	**0.8098**
Rational quadratic ARD	0.6773	0.8238
Matern(5,2)	0.6772	0.8124
Neural network	0.6727	0.8103



  - カーネルを学習できる(ARD)→どの特徴が効くかわかる
- 論文ではさらに、マルチタスク学習でアノテーターの相関と信頼度を自動学習

ガウス過程とカーネルの学習

ハイパーパラメータの最適化

$$k(\mathbf{x}_i, \mathbf{x}_j) = \theta_1 \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{\theta_2}\right) + \theta_3 \delta(i,j)$$

- カーネルのハイパーパラメータを $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ とおくと、yの確率はガウス分布なので、θに依存して

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) &= \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{\boldsymbol{\theta}}) \\ &= \frac{1}{(2\pi)^{N/2}} \frac{1}{|\mathbf{K}_{\boldsymbol{\theta}}|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{y}^T \mathbf{K}_{\boldsymbol{\theta}}^{-1} \mathbf{y}\right) \end{aligned}$$

- すなわち、

$$\log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \log |\mathbf{K}_{\boldsymbol{\theta}}| - \mathbf{y}^T \mathbf{K}_{\boldsymbol{\theta}}^{-1} \mathbf{y} + \text{const.}$$

  - これを最大にするθを求めればよい.

ハイパーパラメータの最適化 (2)

$$L = \log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \log |\mathbf{K}_{\boldsymbol{\theta}}| - \mathbf{y}^T \mathbf{K}_{\boldsymbol{\theta}}^{-1} \mathbf{y} + \text{const.}$$

- ある $\theta \in \boldsymbol{\theta}$ について、微分の連鎖則から

$$\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial \mathbf{K}_{\boldsymbol{\theta}}} \frac{\partial \mathbf{K}_{\boldsymbol{\theta}}}{\partial \theta} = \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\partial L}{\partial K_{ij}} \frac{\partial K_{ij}}{\partial \theta}$$

- ここで

$$\frac{\partial}{\partial \theta} \log |\mathbf{K}_{\boldsymbol{\theta}}| = \text{tr} \left(\mathbf{K}_{\boldsymbol{\theta}}^{-1} \frac{\partial \mathbf{K}_{\boldsymbol{\theta}}}{\partial \theta} \right)$$

$$\frac{\partial}{\partial \theta} \mathbf{K}_{\boldsymbol{\theta}}^{-1} = -\mathbf{K}_{\boldsymbol{\theta}}^{-1} \frac{\partial \mathbf{K}_{\boldsymbol{\theta}}}{\partial \theta} \mathbf{K}_{\boldsymbol{\theta}}^{-1}$$

なので、後は $\frac{\partial K_{ij}}{\partial \theta}$ を使っているカーネルごとに計算すればよい.

ハイパーパラメータの最適化 (3)

- $\dfrac{\partial \mathbf{K}_{\boldsymbol{\theta}}}{\partial \theta}$ は?

  → 各K_{ij}をθで微分して並べた行列.

- 例:

  $$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \theta_1 \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{\theta_2}\right) + \theta_3 \delta(i,j)$$

  のとき、

  θ_1>0 なので　$\theta_1 = e^{\tau} \rightleftarrows \tau = \log \theta_1$　とおけば、

  $$\frac{\partial k(\mathbf{x}_i, \mathbf{x}_j)}{\partial \tau} = e^{\tau} \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{e^{\tau}}\right) = k(\mathbf{x}_i, \mathbf{x}_j) - e^{\tau}\delta(i,j)$$

  – θ_2, θ_3 についても同様

ハイパーパラメータの最適化 (4)

- 最適化アルゴリズム (Python, BFGSの場合)

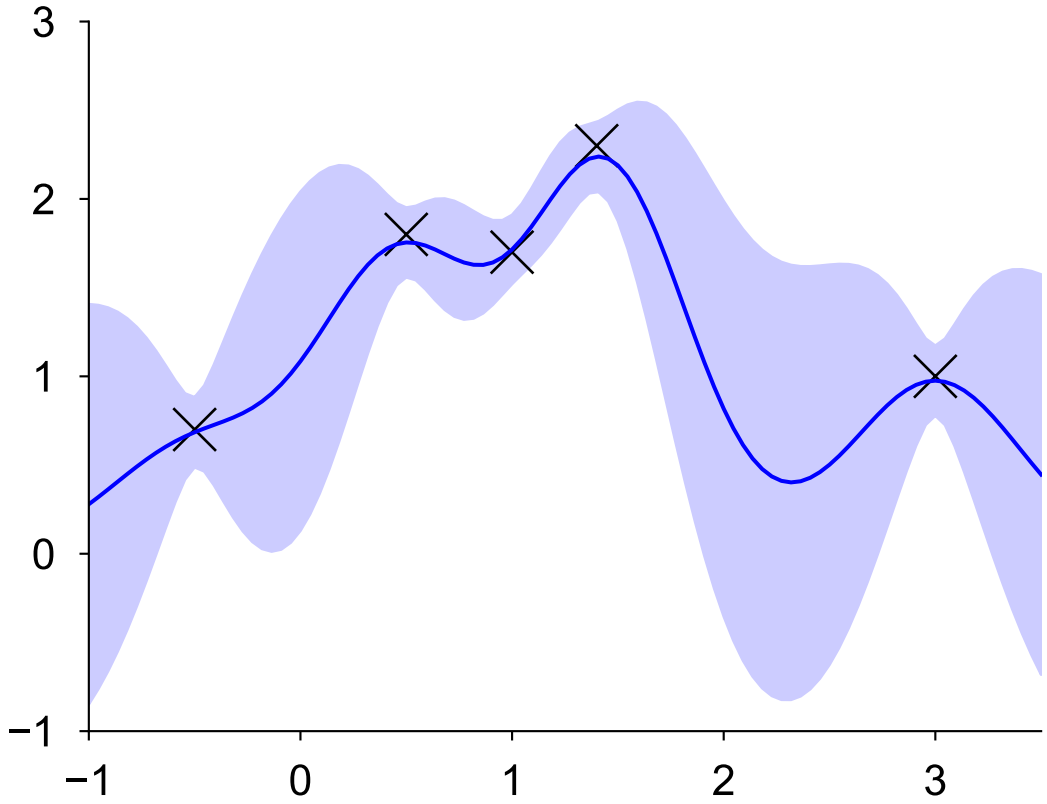
```
from scipy.optimize import minimize

def optimize (xtrain, ytrain, kernel, kgrad, init):
    res = minimize (loglik, init,
                    args = (xtrain,ytrain,kernel,kgrad),
                    jac = gradient, method = 'BFGS',
                    callback = printparam,
                    options = {'gtol' : 1e-4, 'disp' : True})
    print res.message
    return res.x
```

  - loglik で目的関数(負の対数尤度)を、gradientで偏微分を並べたベクトルを計算

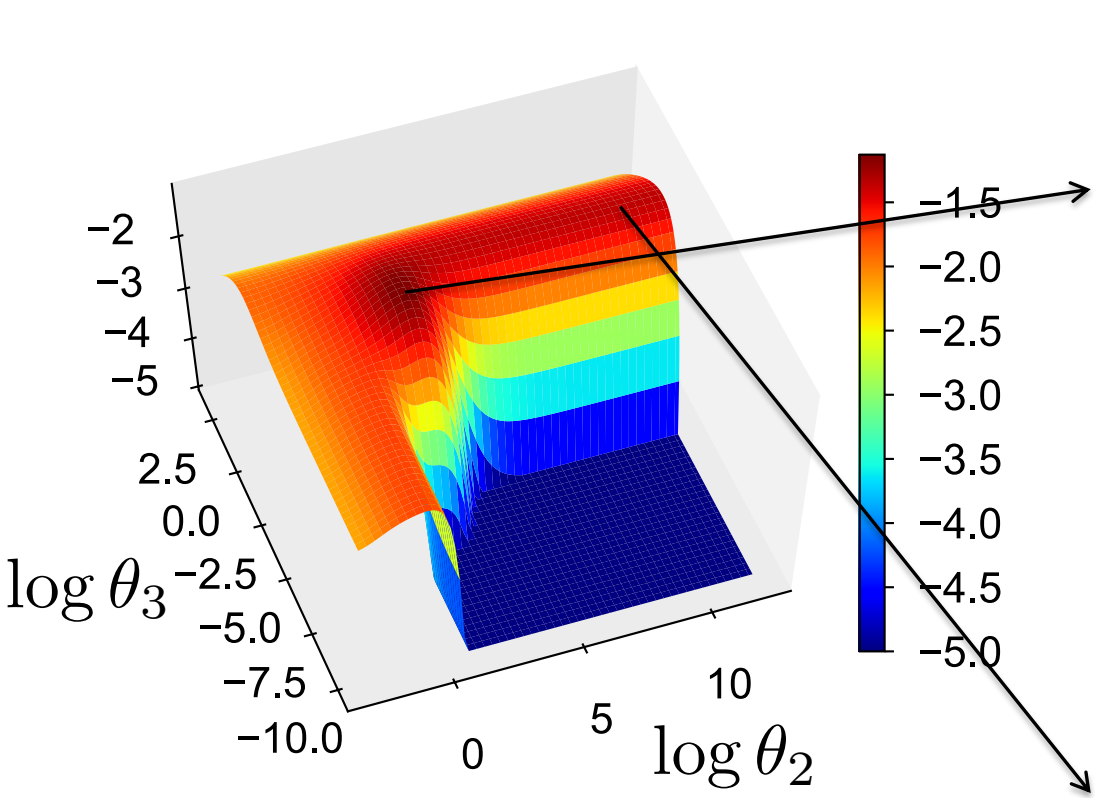
最適化ルーチンは最小化問題を解いているため

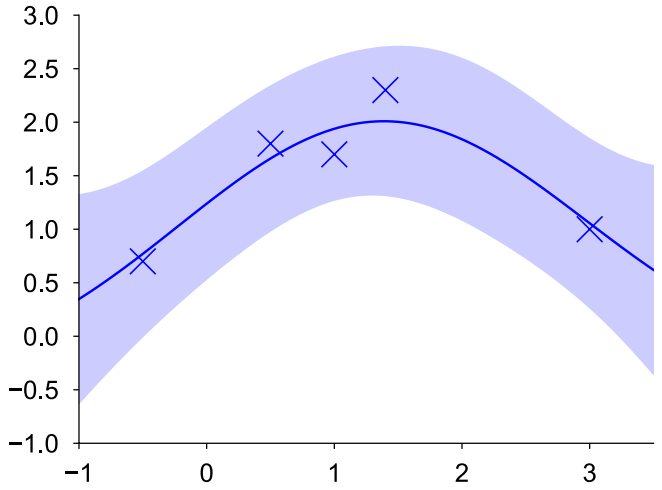
ハイパーパラメータの最適化 (5)

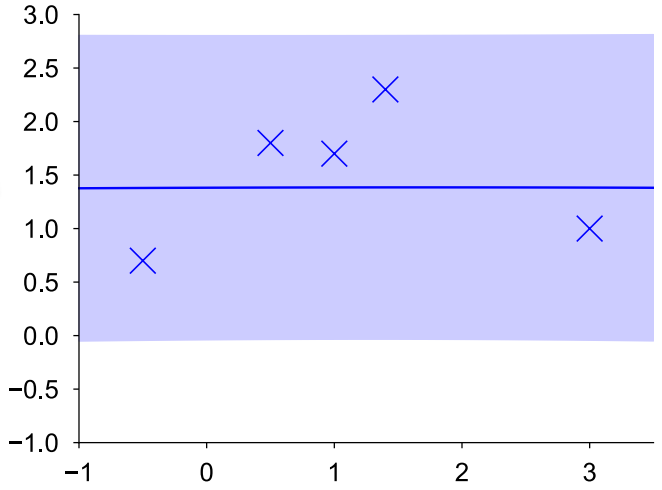


- カーネル $k(\mathbf{x}_i, \mathbf{x}_j) = \theta_1 \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{\theta_2}\right) + \theta_3 \delta(i,j)$ で、$\theta_1 = 1$ としてみる
- 上の画像は、観測点が少ないので若干オーバーフィット

ハイパーパラメータの最適化 (6)







- $\log \theta_2, \log \theta_3$の関数
- 複数の峰があり、最適化は初期値に依存する

カーネルの組み合わせ

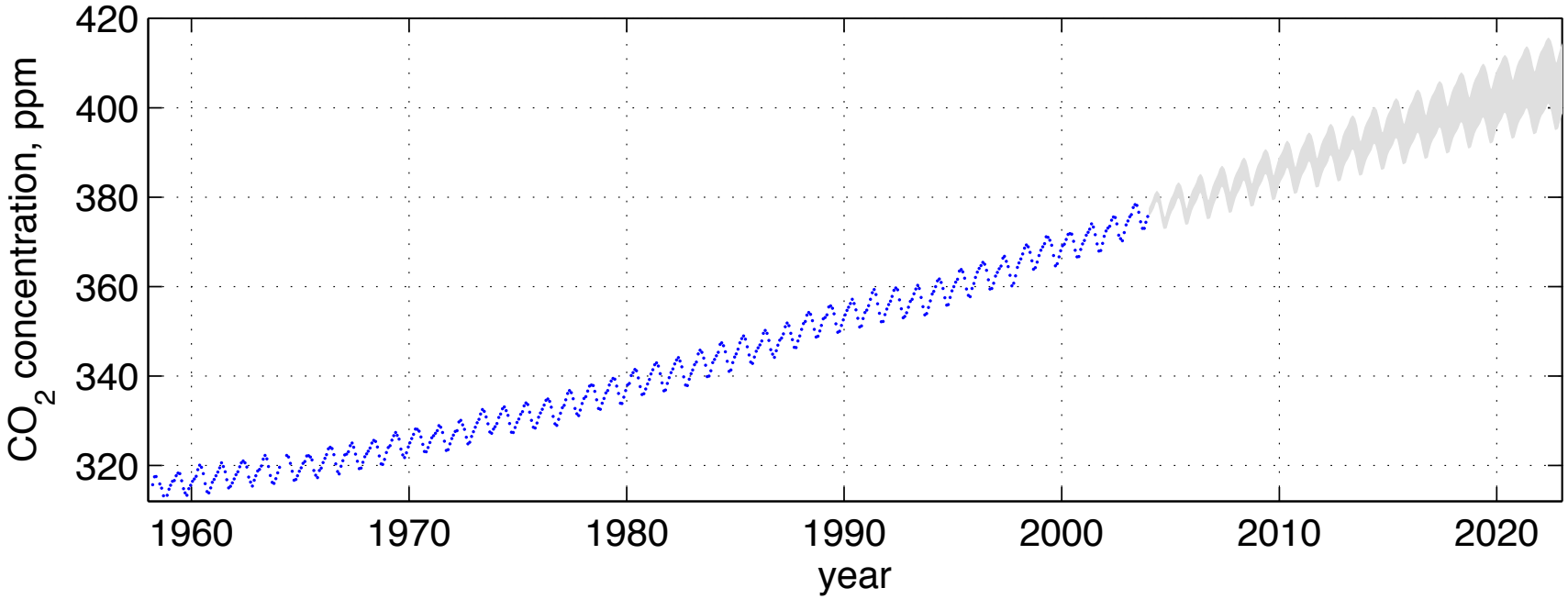
- カーネル $k_1(x,x')$ と $k_2(x,x')$ の和や積も、正しいカーネル関数になる
  - $\theta_1 k_1(\mathbf{x}, \mathbf{x}') + \theta_2 k_2(\mathbf{x}, \mathbf{x}')$
  - $k_1(\mathbf{x}, \mathbf{x}')^p \cdot k_2(\mathbf{x}, \mathbf{x}')^q \quad (p, q \in \mathbb{N})$

  などは、また有効なカーネル関数→GPML4章を参照
- カーネルとして、たとえば

$$k(\mathbf{x}, \mathbf{x}') = \theta_1 \mathbf{x}^T \mathbf{x}' + \theta_2 \exp\left(\theta_3 \cos\left(\frac{|\mathbf{x} - \mathbf{x}'|}{\theta_4}\right)\right) \quad (\theta_1, \theta_2, \theta_3, \theta_4 \geq 0)$$

を使って $\theta_1, \theta_2, \theta_3, \theta_4$ を最適化すれば、線形性と周期性を自動的に調節した回帰モデルが得られる!

マウナロアCO_2濃度データ



- GPML p.119より引用
- 周期カーネルを使うことで、非常に正確に周期データにフィットして予測することができる!

ガウス分布以外の観測モデル (離散値を含む)

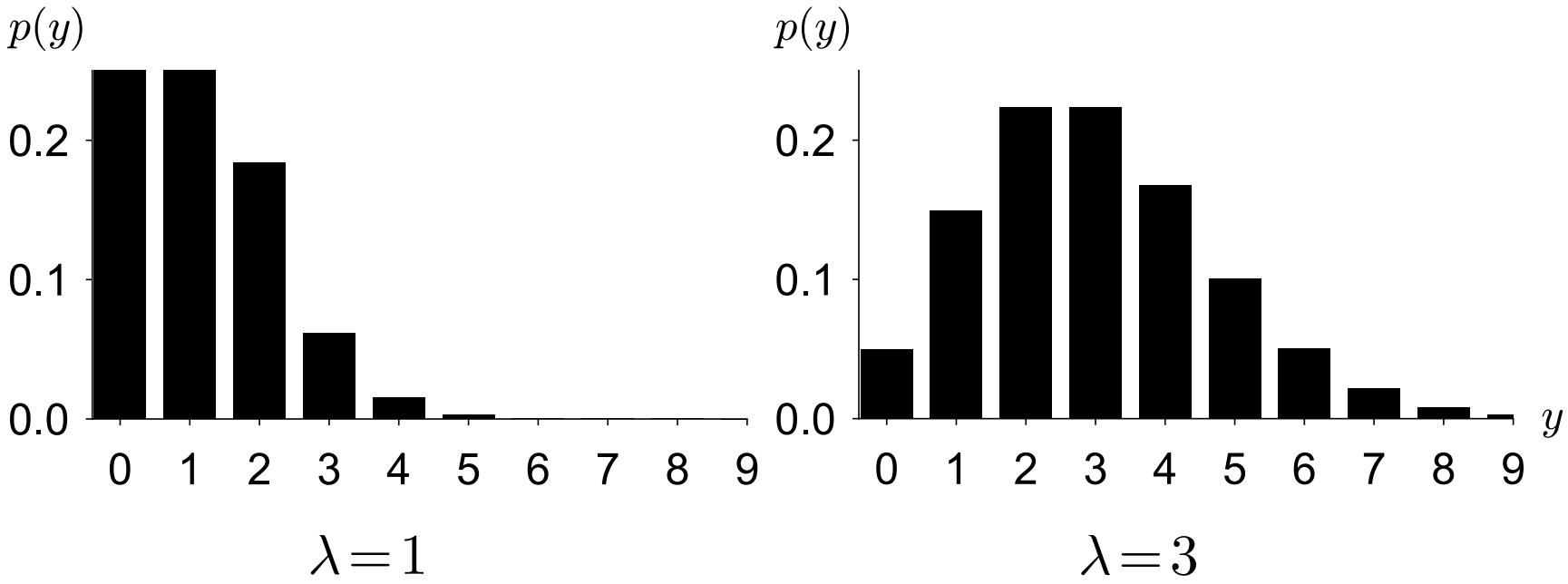
さまざまな観測モデル

- 論文では $p(\mathbf{y}|\mathbf{f})$ がガウス分布だと仮定されることが多いが、現実の観測値yはガウス分布とは限らない
  - 離散観測値：y=2,1,4,3,1,0,1,...
  - 外れ値の存在 (ガウス分布では外れ値を扱えない)
  - 点過程データ：イベントがfに基づいてランダムに生起

ポアソン観測モデル

- ポアソン分布: 自然数上の確率分布

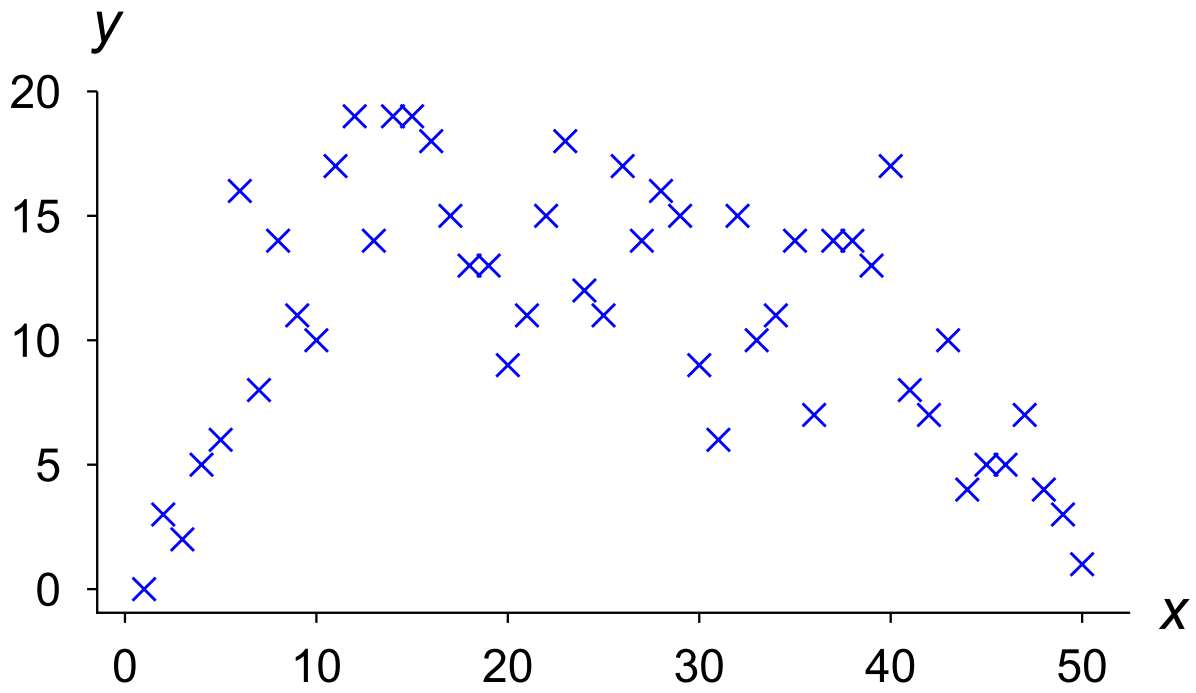
$$p(y|\lambda) = \frac{\lambda^y e^{-\lambda}}{y!} \quad (y = 0, 1, 2, \cdots)$$



ポアソン観測モデル (2)

$$p(y|\lambda) = \frac{\lambda^y e^{-\lambda}}{y!} \quad (y = 0, 1, 2, \cdots)$$

- $\lambda > 0$ なので、$\lambda = e^{f(\mathbf{x})}$ とおく
(場所 $\mathbf{x}$ におけるポアソン分布の期待値)



場所xにおける植物の個体数の架空データ
(久保拓弥「データ解析のための統計モデリング入門」より)

ポアソン観測モデル (3)

- ここでも、事後分布はガウス分布ではない

$$p(\mathbf{f}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{f})p(\mathbf{f})$$
$$= \prod_{n=1}^{N} \frac{\exp(f(x_n)y_n - e^{f(x_n)})}{y_n!} \cdot \exp\left(-\frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f}\right)$$

- たとえば`GPy`で計算

ポアソン観測モデル (4)

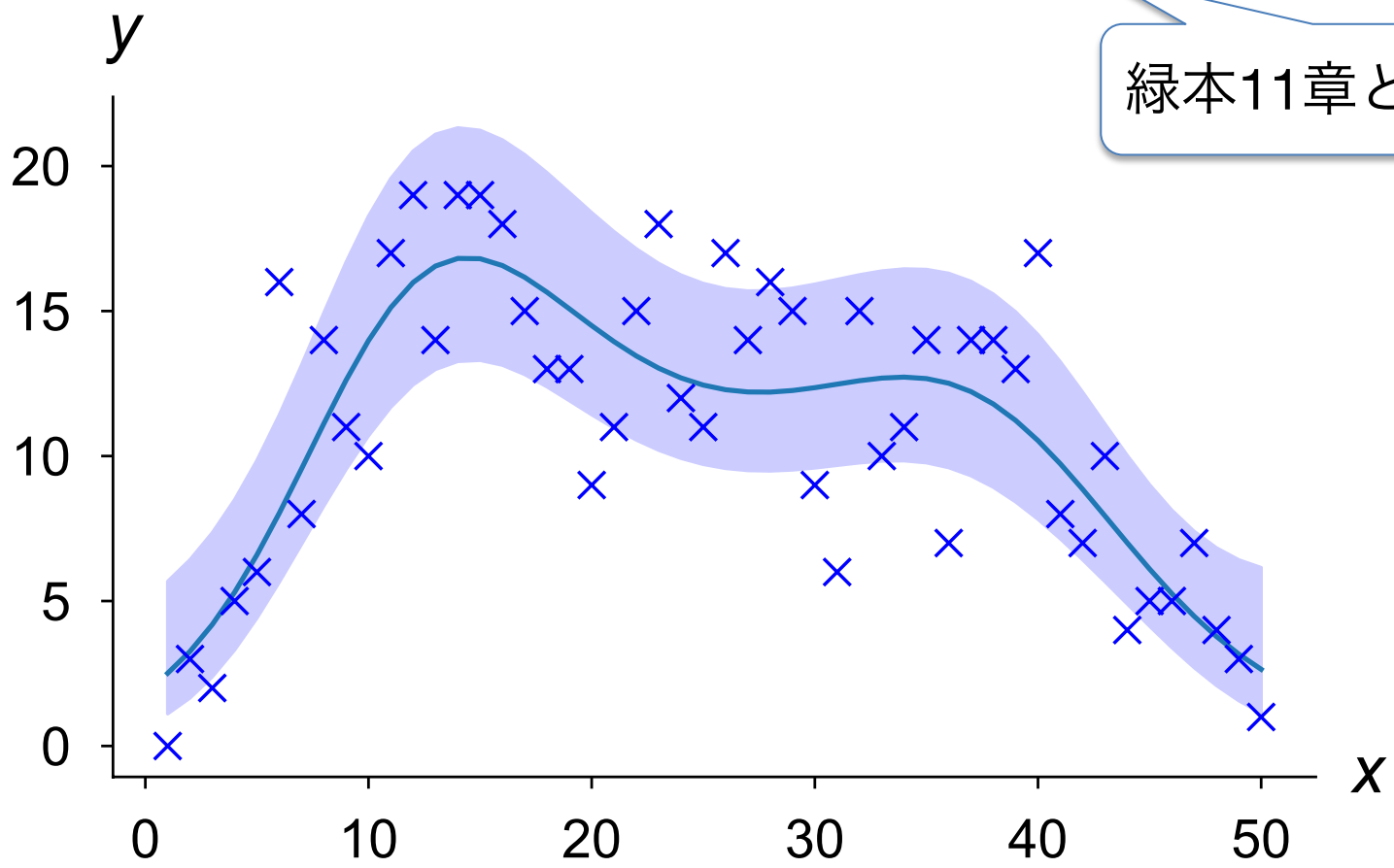
- たとえばGPyで計算

```
import GPy

def gpr_poisson (data):
    N = len(data)
    xx = np.linspace (1,N,N)
    model = GPy.core.GP (X=xx[:,None], Y=data[:,None],¥
                         kernel=GPy.kern.RBF(1),¥
                         inference_method=GPy.inference.latent_
                         function_inference.Laplace(),¥
                         likelihood=GPy.likelihoods.Poisson())
    model.optimize ()
    mu,var = model._raw_predict (xx[:,None])
    plt.plot (xx, np.exp(mu))
    plt.fill_between (xx, exp(mu[:,0] + 3*sqrt(var[:,0])),¥
                      exp(mu[:,0] - 3*sqrt(var[:,0])),¥
                      color='#ccccff')
    plt.plot (xx, data, 'xb', markersize=8)
```

ポアソン観測モデル (5)

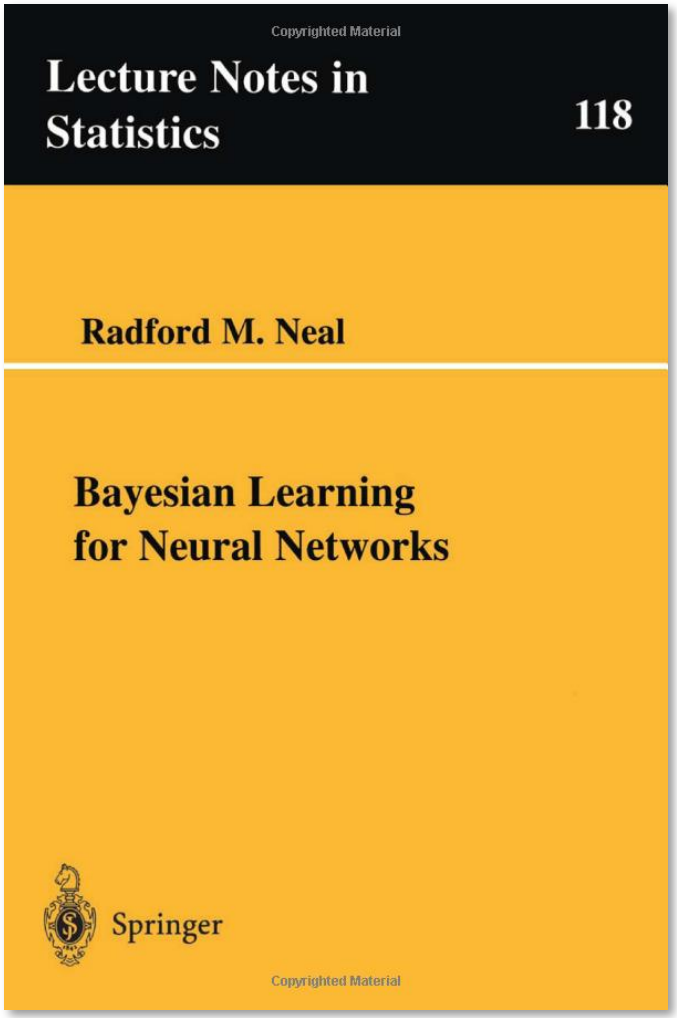
- 推定した結果 (ラプラス近似; 分散が小さめ)
  - 空間的に滑らかな推定結果が得られる



緑本11章と比較！

ガウス過程とニューラルネット

ニューラルネットとガウス過程

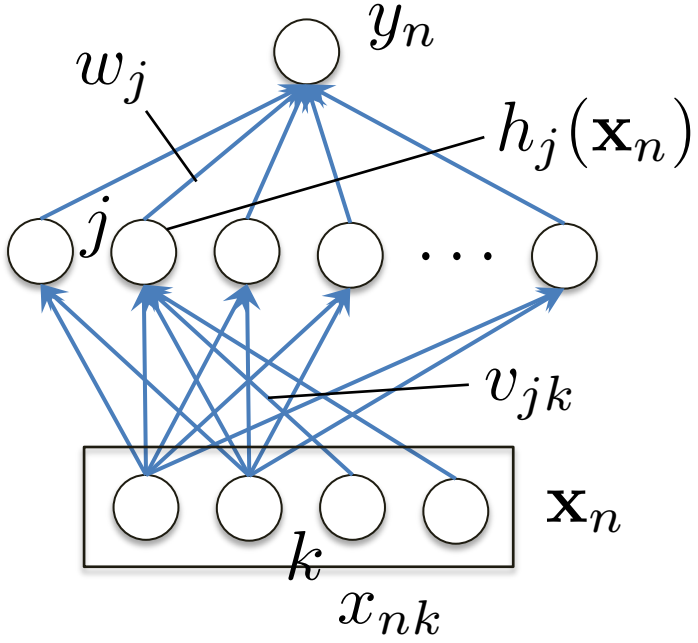


- Neal (1996)は、ニューラルネットワークは素子数→∞の極限でガウス過程と等価であることを示した
- ニューラルネットの多数のパラメータを学習する必要がない!
- 以下、その説明を簡単に紹介

ニューラルネットとガウス過程 (2)

- 入力 $\mathbf{x}_n$ に対して y_n を出力する、右図のような1層のニューラルネットを考える
- 式で書くと、

$$\begin{cases} y_n = \sum_{j=1}^{H} w_j h_j(\mathbf{x}_n) \\ h_j(\mathbf{x}_n) = \sigma(\sum_{k=0}^{D} v_{jk} x_{nk}) \end{cases}$$



  - ただし重み w, v はi.i.d.に

$$w_j \sim \mathcal{N}(0, \sigma_w^2),\ v_{jk} \sim \mathcal{N}(0, \sigma_v^2/H)$$

に従うとする

ニューラルネットとガウス過程 (3)

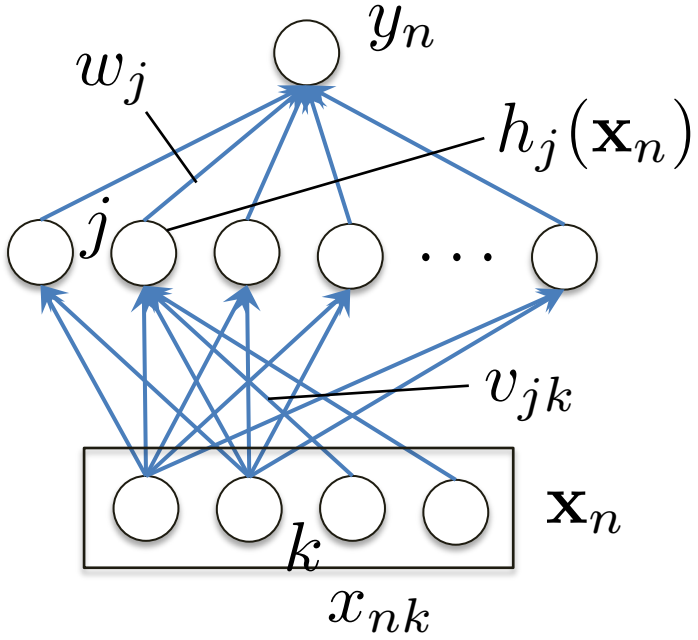
- このとき、y_n の期待値は?
- まず、

$$E[h_j(\mathbf{x}_n)] = E[\sigma(\sum_{k=0}^{D} v_{jk} x_{nk})]$$

  は j に関わらず同じ分布
- よって、中心極限定理より

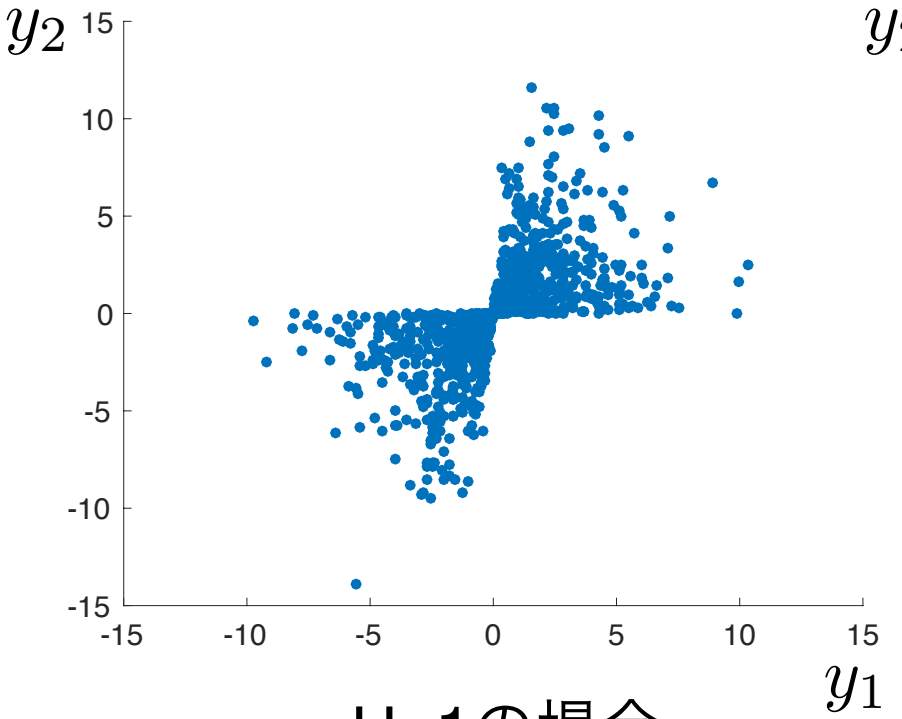
$$y_n = \sum_{j=1}^{H} w_j h_j(\mathbf{x}_n)$$

  は、$H \to \infty$ で平均0のガウス分布に収束
- 共分散 $V[y_n y_m] = E[y_n y_m]$ の計算も同様
  → $\mathbf{y} = (y_1, y_2, \ldots, y_N)$ は多変量ガウス分布に収束
  - 詳しい計算は、Neal (1996)を参照のこと

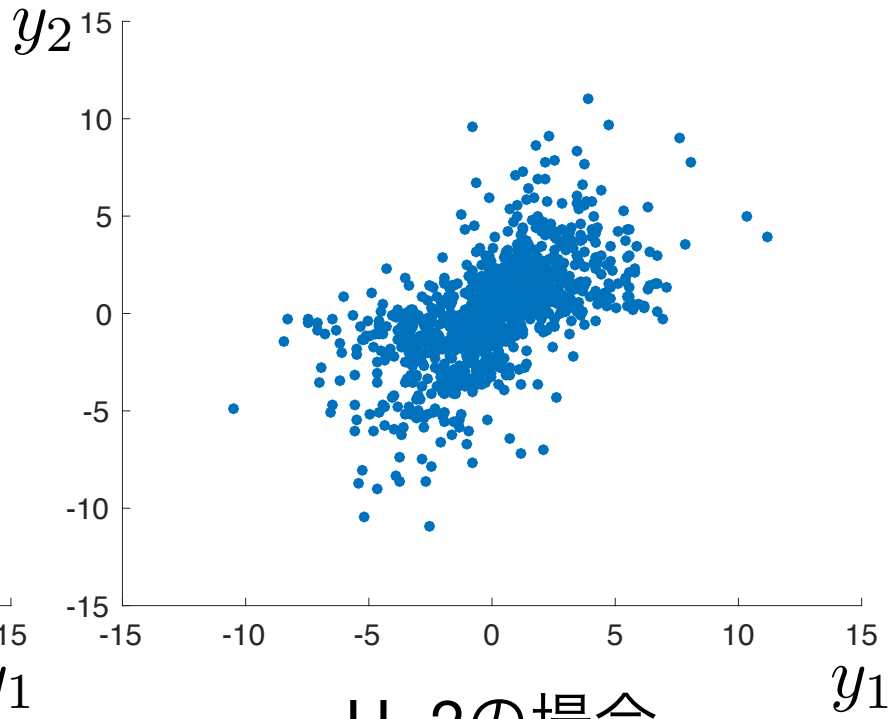


ニューラルネットとガウス過程 (4)

- 事前分布からランダムに生成したニューラルネットについて、入力 $(x_1, x_2) = (-0.2, 0.4)$ に対する出力 (y_1, y_2) をプロット

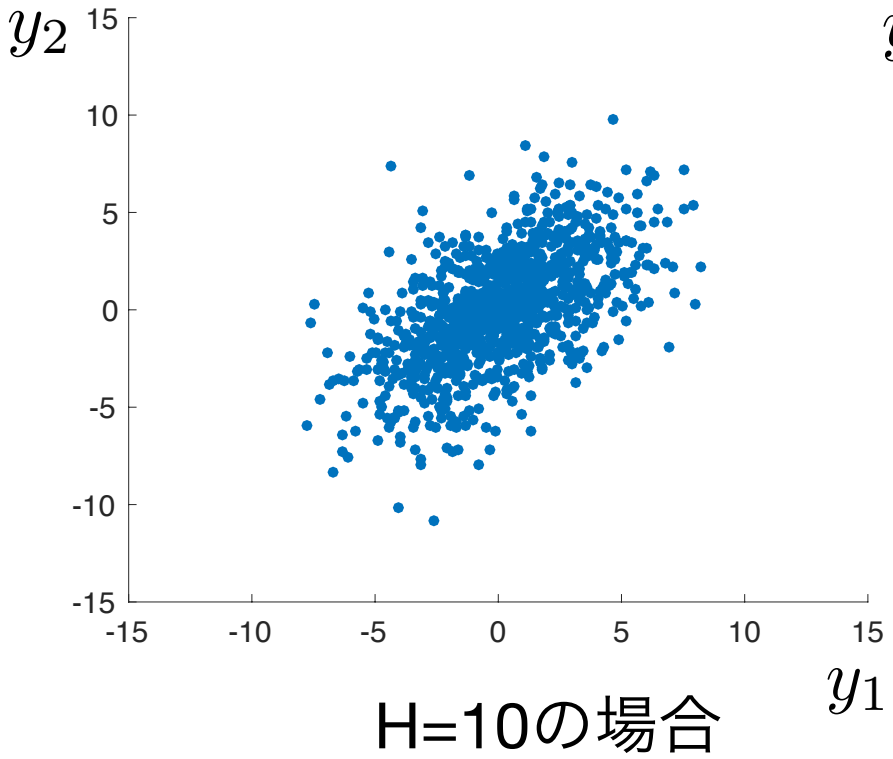


H=1の場合

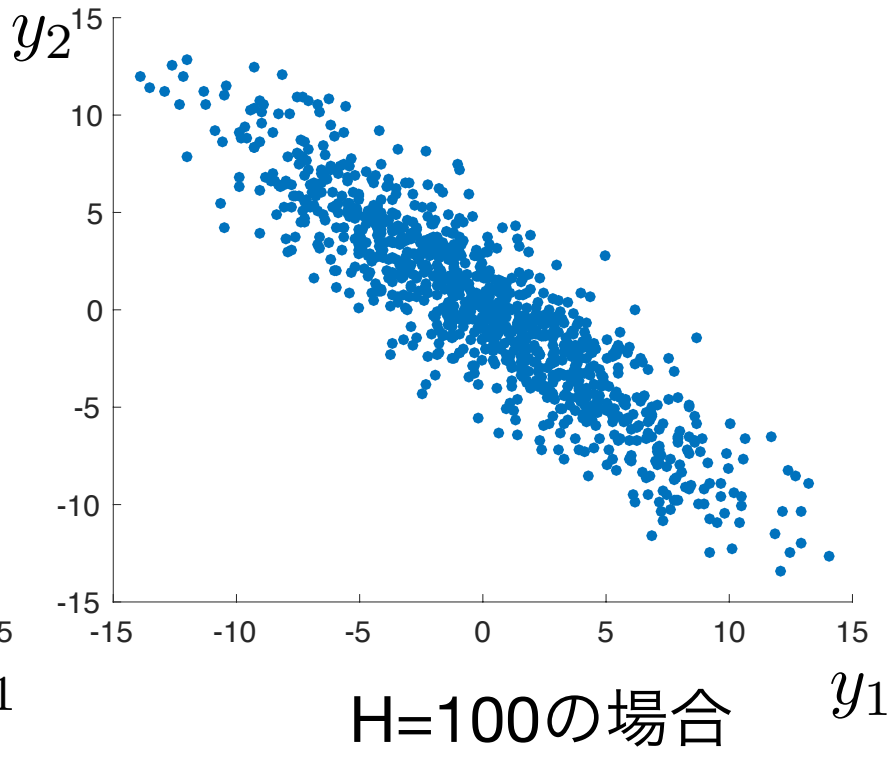


H=2の場合

ニューラルネットとガウス過程 (5)



H=10の場合



H=100の場合

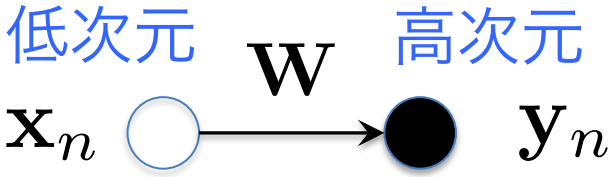
- (y_1,y_2)の同時分布は多変量ガウス分布に漸近する！
  - ノード数H=10ですでにほぼガウス分布と等価
  - 中心極限定理の効果

ガウス過程による教師なし学習

ガウス過程と教師なし学習

- ガウス過程回帰モデルでは、入力xと出力yのペア$(\mathbf{x}, y)$が与えられていた
- 観測値 $\mathbf{y}$ しかない場合はどうする?
- 非常によくある設定 (教師なし学習)
  - y＝あるユーザーのクリック履歴
  - y＝ロボットの姿勢ベクトル (各関節角のベクトル)
  - y＝ある星の吸収線スペクトル
  - ここでは、yが連続値の場合を考える

確率的主成分分析



- Probabilistic PCA (Tipping & Bishop 1999)

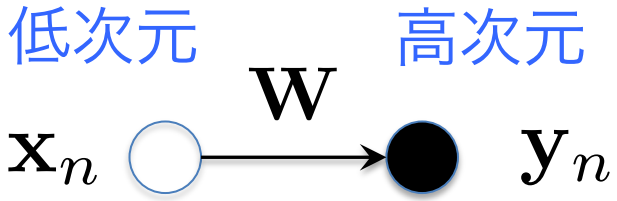
$$\begin{cases} \mathbf{y}_n = \mathbf{W}\mathbf{x}_n + \boldsymbol{\epsilon} \\ \boldsymbol{\epsilon} \ \sim \mathrm{N}(\mathbf{0}, \sigma^2\mathbf{I}) \end{cases}$$

- よって、$L = \sum_{n=1}^{N} \log p(\mathbf{y}_n) = \sum_{n=1}^{N} \log \mathcal{N}(\mathbf{y}_n | \mathbf{W}\mathbf{x}_n, \sigma^2\mathbf{I})$

$$= -\frac{N}{2}\left(\log 2\pi + \log|C| + \mathrm{tr}(C^{-1}S)\right)$$

$$(C = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I},\ S = \mathbf{Y}\mathbf{Y}^T/N)$$

確率的主成分分析 (2)



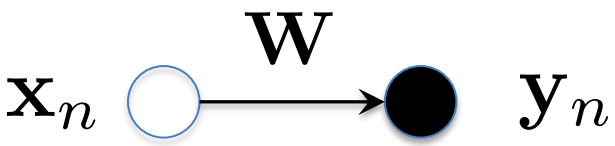
- $\partial L/\partial \mathbf{W} = 0$ より、データの尤度Lを最大にするWの最尤推定値は

$$\mathbf{W} = \mathbf{U}_q(\mathbf{\Lambda}_q - \sigma^2\mathbf{I})^{1/2}$$

  - $\mathbf{\Lambda}_q, \mathbf{U}_q$： $\mathbf{YY}^T$ の最大q個の固有値・固有ベクトルを並べた行列
  - $\sigma^2 = 0$ で通常の主成分分析と一致

確率的PCAからGPLVMへ

- 確率的PCA：x→yへの射影行列Wを最適化



- Wは巨大、xの次元に依存する
  → Wの方に事前分布を与えて積分消去

$$p(\mathbf{W}) = \prod_{d=1}^{D} \mathrm{N}(\mathbf{w}_d|\mathbf{0}, \alpha^{-1}\mathbf{I})$$

$$p(\mathbf{Y}|\mathbf{X}) = \int p(\mathbf{Y}|\mathbf{X}, \mathbf{W})p(\mathbf{W})d\mathbf{W}$$

$$= \frac{1}{(2\pi)^{DN/2}|\mathbf{K}|^{D/2}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T)\right)$$

GPLVM

- よって、

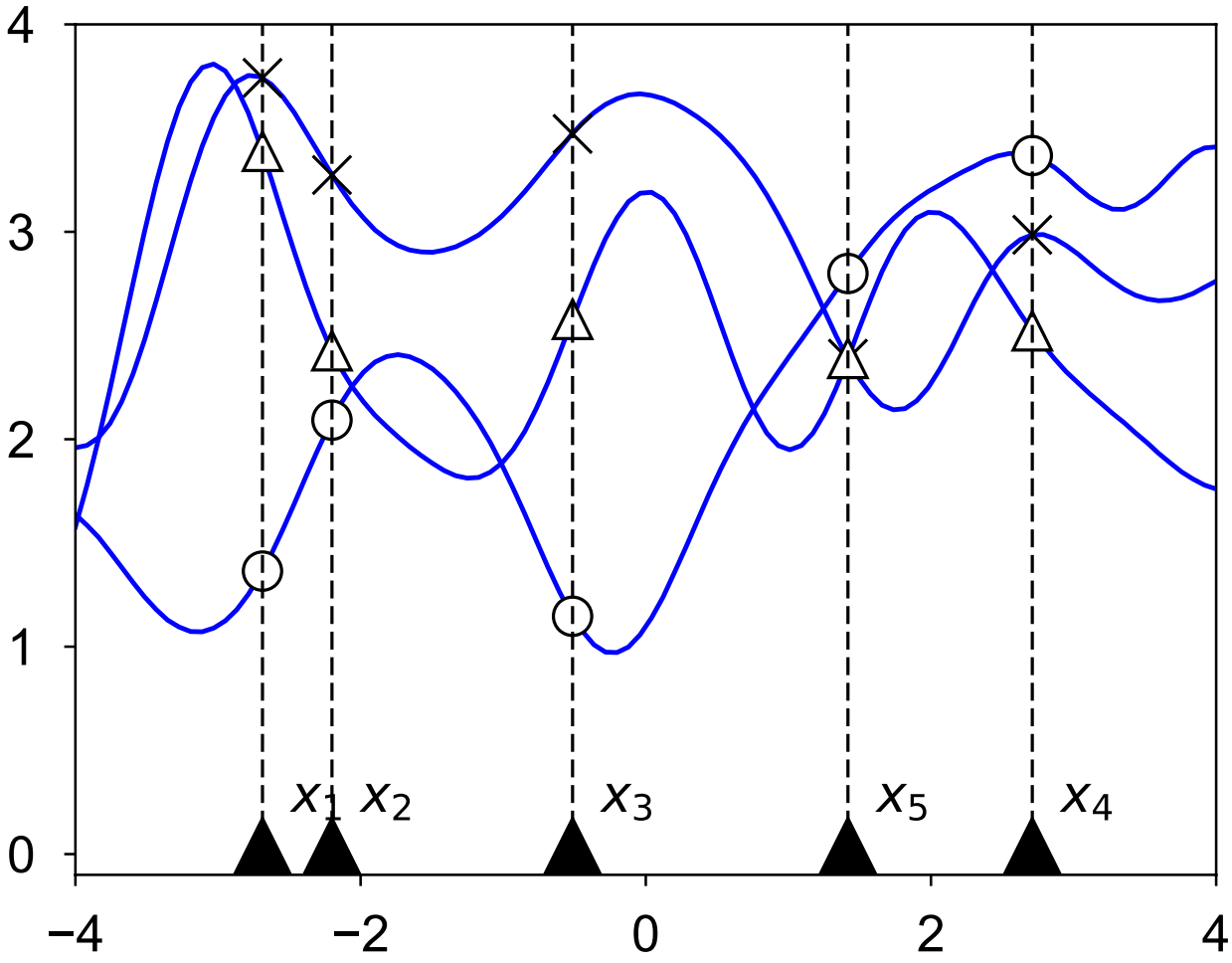
$$\log p(\mathbf{Y}|\mathbf{X}) = -\frac{DN}{2}\log(2\pi) - \frac{D}{2}\log|\mathbf{K}_\mathbf{X}| - \frac{1}{2}\mathrm{tr}(\mathbf{K}_\mathbf{X}^{-1}\mathbf{Y}\mathbf{Y}^T)$$

$$\mathbf{K}_\mathbf{X} = \alpha\mathbf{X}\mathbf{X}^T + \beta^{-1}\mathbf{I}$$

$$\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_N)$$

- これを最大化する潜在的なXを見つければよい
- Gaussian Process Latent Variable Model (GPLVM)という (Lawrence+, NIPS 2003)

GPLVMのイメージ



- xが1次元の場合: 各観測値 (×,△,○)の背後にxが存在

GPLVMの最適化

$$\log p(\mathbf{Y}|\mathbf{X}) = -\frac{DN}{2}\log(2\pi) - \frac{D}{2}\log|\mathbf{K}_\mathbf{X}| - \frac{1}{2}\mathrm{tr}(\mathbf{K}_\mathbf{X}^{-1}\mathbf{Y}\mathbf{Y}^T)$$

$$\mathbf{K}_\mathbf{X} = \alpha\mathbf{X}\mathbf{X}^T + \beta^{-1}\mathbf{I}$$

$$\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_N)$$

- 自然にカーネル化されている→任意のカーネルを導入

$$k(\mathbf{x}_n, \mathbf{x}_m) = \alpha\exp\left(-\gamma|\mathbf{x}_n - \mathbf{x}_m|^2\right) + \delta(n,m)\beta^{-1}$$

(RBF)

- $\dfrac{\partial L}{\partial \mathbf{K}_\mathbf{X}} = \mathbf{K}_\mathbf{X}^{-1}\mathbf{Y}\mathbf{Y}^T\mathbf{K}_\mathbf{X}^{-1} - D\,\mathbf{K}_\mathbf{X}^{-1}$
  - $\dfrac{\partial L}{\partial x_{nj}} = \dfrac{\partial L}{\partial \mathbf{K}_\mathbf{x}}\dfrac{\partial \mathbf{K}_\mathbf{X}}{\partial x_{nj}}$ を適用して微分

GPLVMの最適化 (2)

- Python実装：『ガウス過程と機械学習』サポートページ
  - http://chasen.org/~daiti-m/gpbook/
- Neil Lawrence によるMATLAB原実装
  - http://inverseprobability.com/gplvm/

GPLVM：計算例

- Oil flowデータ (PRML掲載と同じもの)

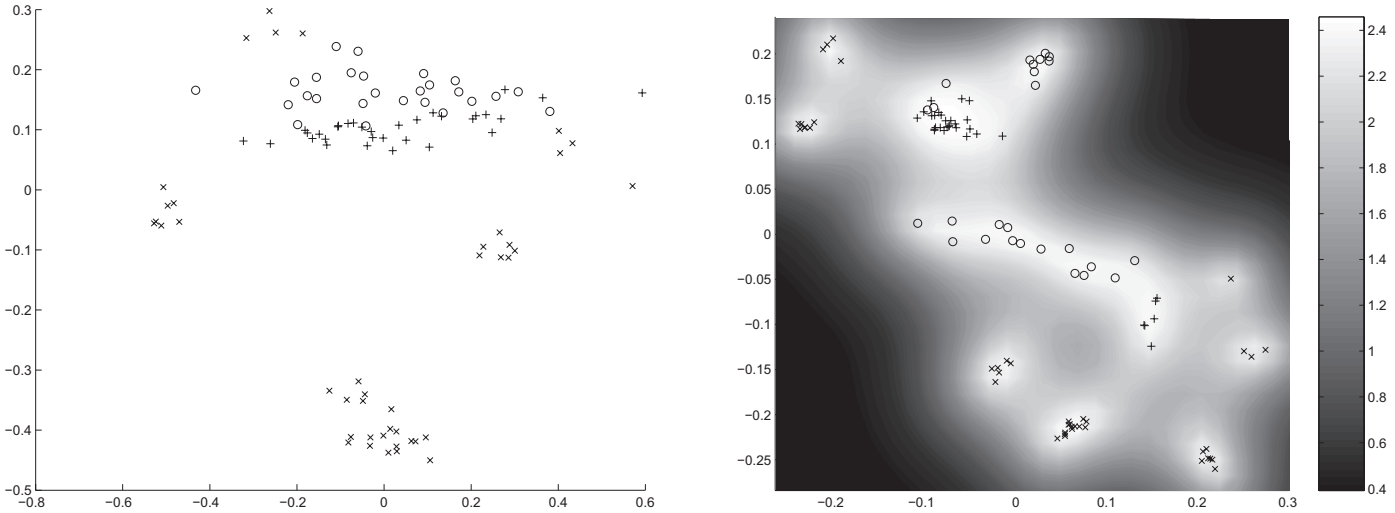
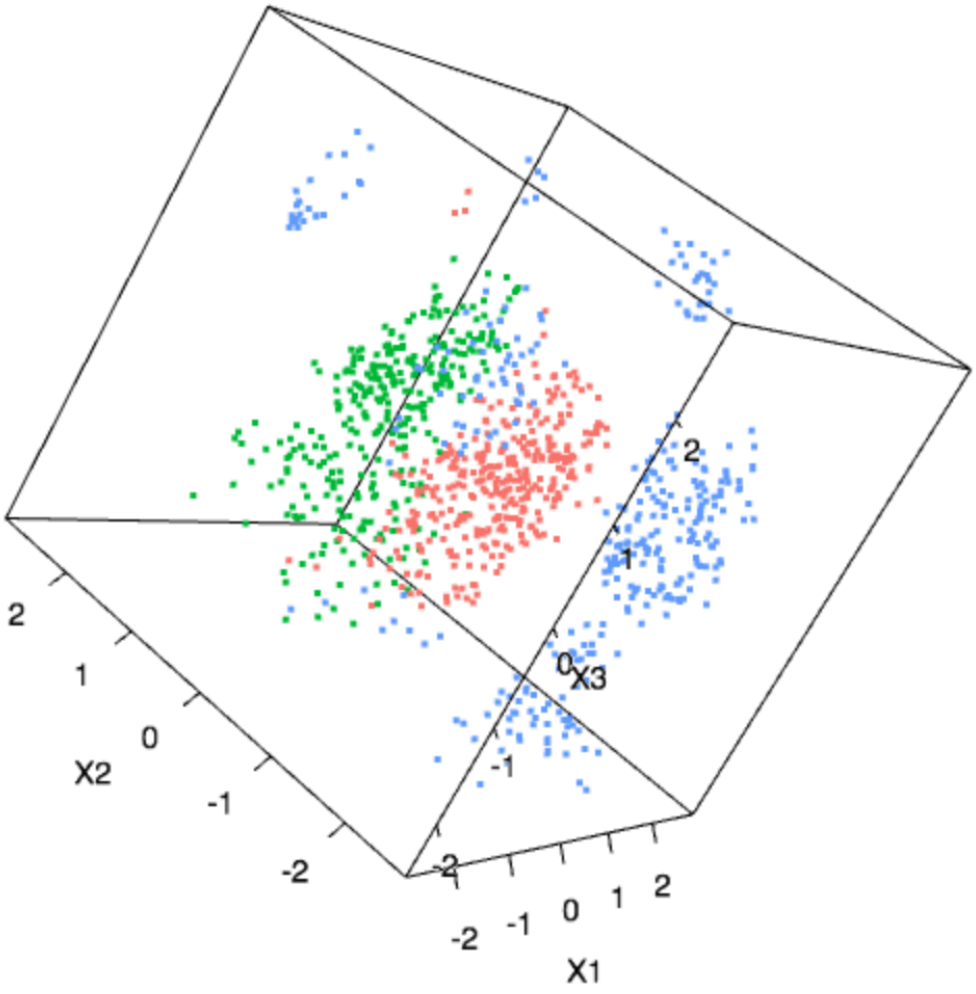


Figure 1: Visualisation of the Oil data with (a) PCA (a linear GPLVM) and (b) A GPLVM which uses an RBF kernel. Crosses, circles and plus signs represent stratifi ed, annular and homogeneous fbws respectively. The greyscales in plot (b) indicate the precision with which the manifold was expressed in data-space for that latent point. The optimised parameters of the kernel were $\gamma = 150$, $\alpha = 0.403$ and $\beta = 316$.

- 左: 確率的PCA、右: GPLVM
- GPLVMは分散を使ってconfidenceの分布も得られる

GPLVM: 3次元の場合

- 松浦さん “Statmodeling memorandum” http://statmodeling.hatenablog.com/entry/gaussian-process-latent-variable-model-2 による
- Stan言語による推定



ガウス過程の自然言語処理への応用

スペクトル混合カーネル

- “Gaussian Process Kernels for Pattern Discovery and Extrapolation”, Andrew Gordon Wilson, Ryan Prescott Adams, ICML 2013.
- ガウス過程で使うカーネルを、RBFのような既存のカーネルおよびその組み合わせに限定せず、フーリエ領域で混合ガウス分布を考えることでデータから自動的に学習できる (!)

スペクトル混合カーネル (2)

- ガウス過程のカーネルとして、値が $\tau = x - x'$ だけに依存する、定常カーネル*k*(τ)を考える
  - RBF (ガウス)カーネルもこの仲間
- ボホナーの定理により、任意の*k*(τ)は

$$k(\tau) = \int_{\mathbb{R}^D} e^{2\pi i\, s^T \tau} \psi(ds)$$

の形に表せる (逆フーリエ変換)

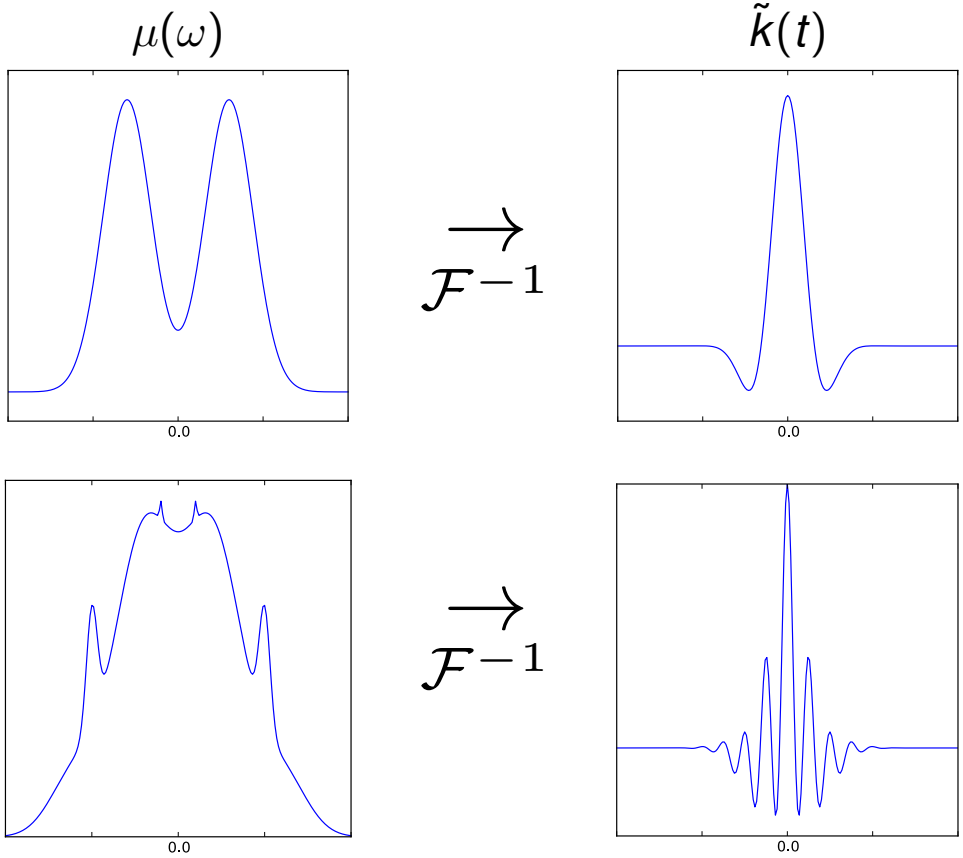
- $\psi(s)$ が、周波数領域での*k*(τ)の等価な表現

↓

k(τ)を確率分布で表せる！

スペクトル混合カーネルのイメージ

周波数空間
(確率分布)



カーネル
(関数)

Gaussian process summer school 2013の資料
“Kernel Design” (N. Durrande) より引用

スペクトル混合カーネル (3)

- 通常のガウスカーネル

$$k(x, x') = \exp\left(-\frac{1}{2}(x - x')^2/\ell^2\right)$$

の周波数表現は、フーリエ変換すると

$$S(s) = (2\pi\ell^2)^{1/2} \exp(-2\pi^2\ell^2 s^2)$$

  - 中心0のガウス分布!

スペクトル混合カーネル (4)

- k(τ)は周波数領域での確率密度 $\psi(s)$ と等価なので、$\psi(s)$ に関して混合ガウス分布を考える
  - 0に関して対称なので、正だけ考えて鏡映

$$\phi(s \,|\, \mu, \sigma^2) = \mathcal{N}(s \,|\, \mu, \sigma^2)$$
$$S(s) = (\phi(s) + \phi(-s))/\,2$$

- ガウス分布の各要素は、もとの領域では以下のカーネル関数を考えていることと等価

$$k(\tau|\sigma, \mu) = \exp(-2\pi^2\sigma^2\tau^2)\cos(2\pi\mu\tau)$$

スペクトル混合カーネル (5)

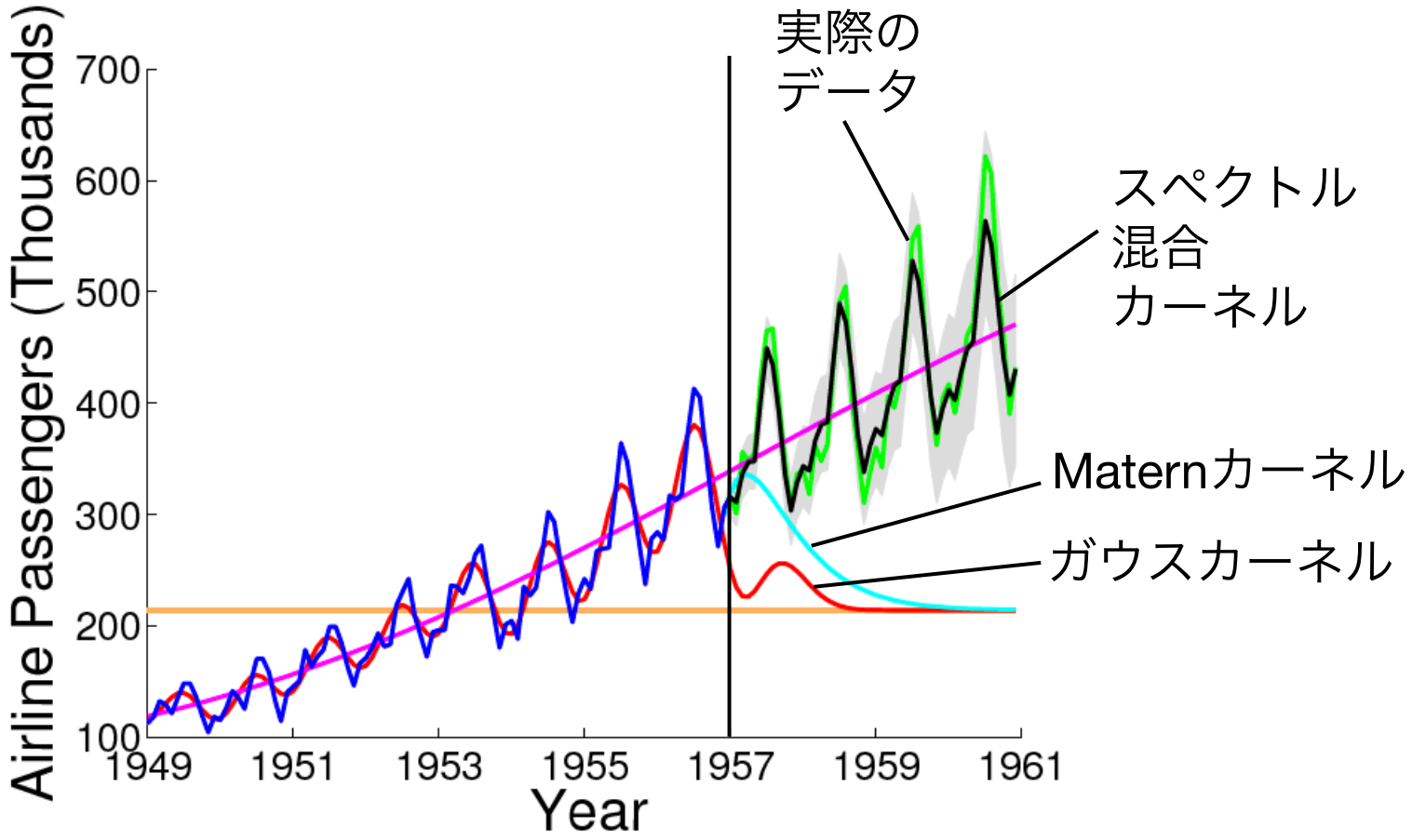
- すなわち、提案法はカーネルとして、次の混合を考えていることになる (Spectral Mixture kernel)

$$k(\boldsymbol{\tau}) = \sum_{p=1}^{P} w_p \cos(2\pi \boldsymbol{\tau}^T \boldsymbol{\mu}^{(p)}) \exp\left(-\sum_{d=1}^{D} 2\pi^2 \sigma_d^{(p)} \tau_d^2\right)$$

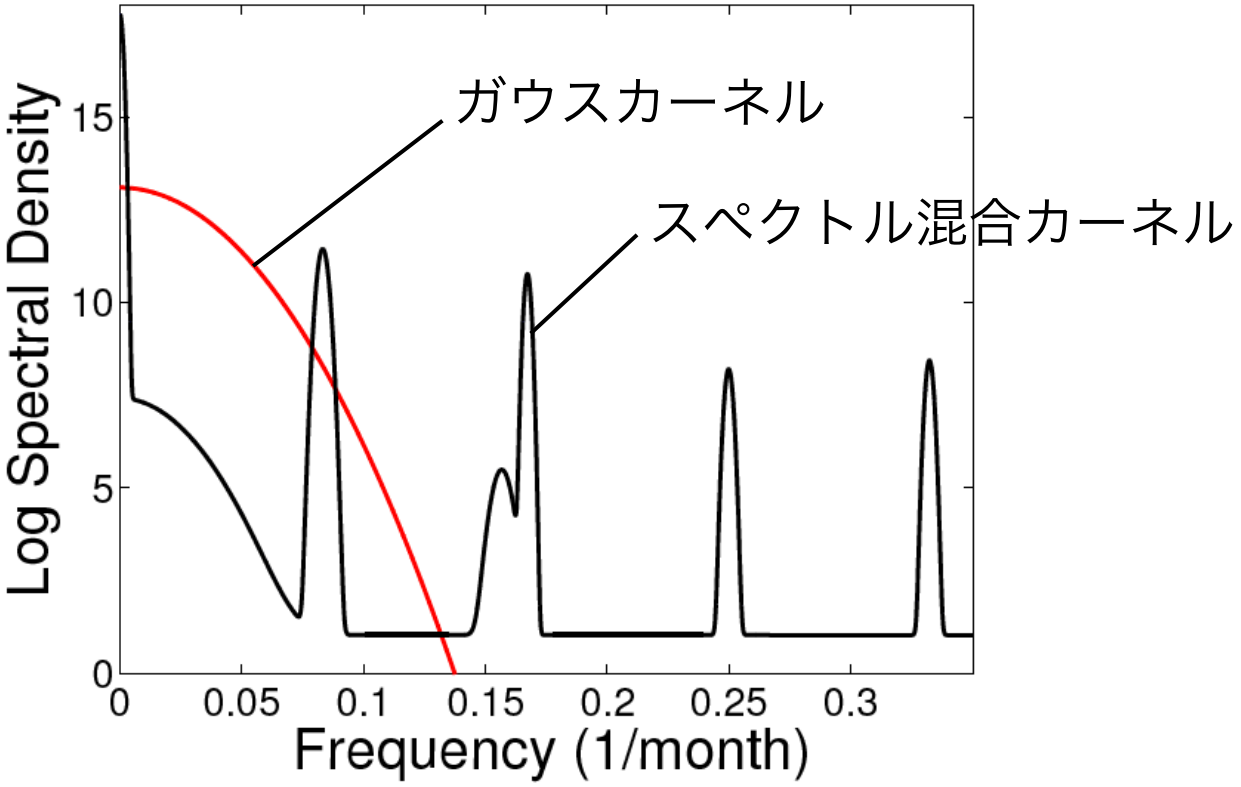
  - パラメータw、μ、σは通常のハイパーパラメータ最適化で学習できる
  - ARD事前分布を使うことで、不要なガウス分布を除去している

Airline Passengerデータ

- 1949-1961の毎月の航空乗客数のうち、最初の8年を学習に使って残りの4年分を予測



Airline Passengerデータ (2)



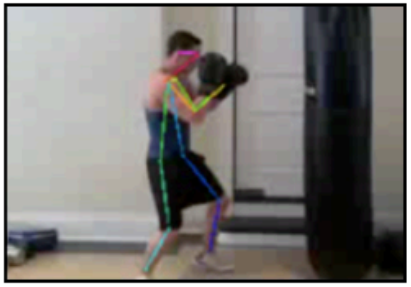
- 線形トレンド(一番左)に加え、12, 6, 4, 3ヶ月の周期性とその細かな違いを捉えている
  - 通常のガウスカーネル(赤)はまったくの大雑把

人間の動作からの副詞の理解

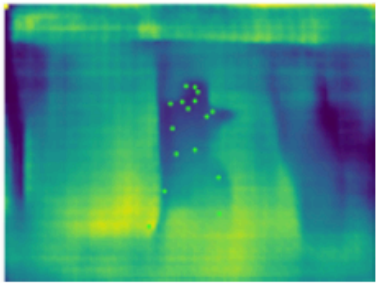
- 『スペクトル混合カーネルとガウス過程に基づく動画からの副詞の意味理解』(P6-17, 3/17 15:20〜)
  - 谷口巴(お茶大)、持橋大地(統数研)、他
- “しっかりと”、“さっと”、“慎重に”といった副詞の理解は、特に介護などで今後とても重要
  - これらは静的な画像ではなく、動作の動的な時系列の性質に関連している
  - 適当なニューラル手法では、動的な性質を十分捉えることができない

動画からの副詞の理解

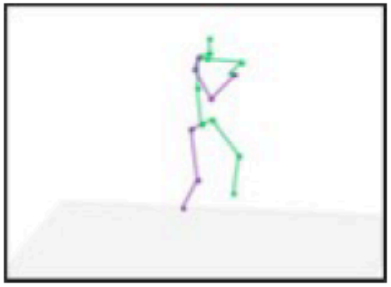
- 動作を表す動画に、「そわそわ」「速足で」「こっそりと」など副詞をクラウドソーシングで
  タグ付けしてもらう
- 動画からOpenPose+前処理で骨格を抽出



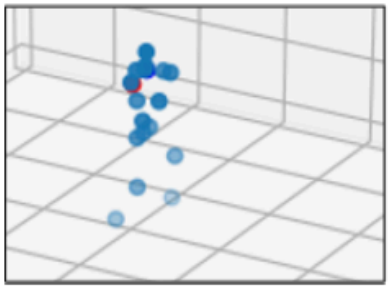
(a) Openpose による画面座標推定



(b) FCRN-depth による深度推定



(c) 3 次元の骨格座標の推定

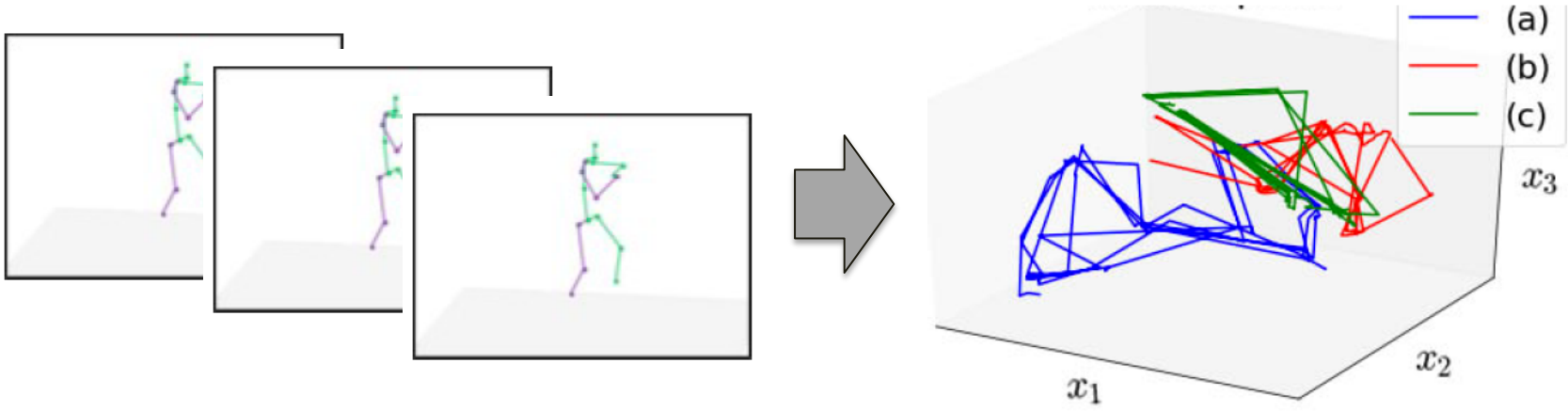


(d) 回転行列による方向正規化

- 48次元の関節座標の時系列データが得られる

動画からの副詞の理解 (2)

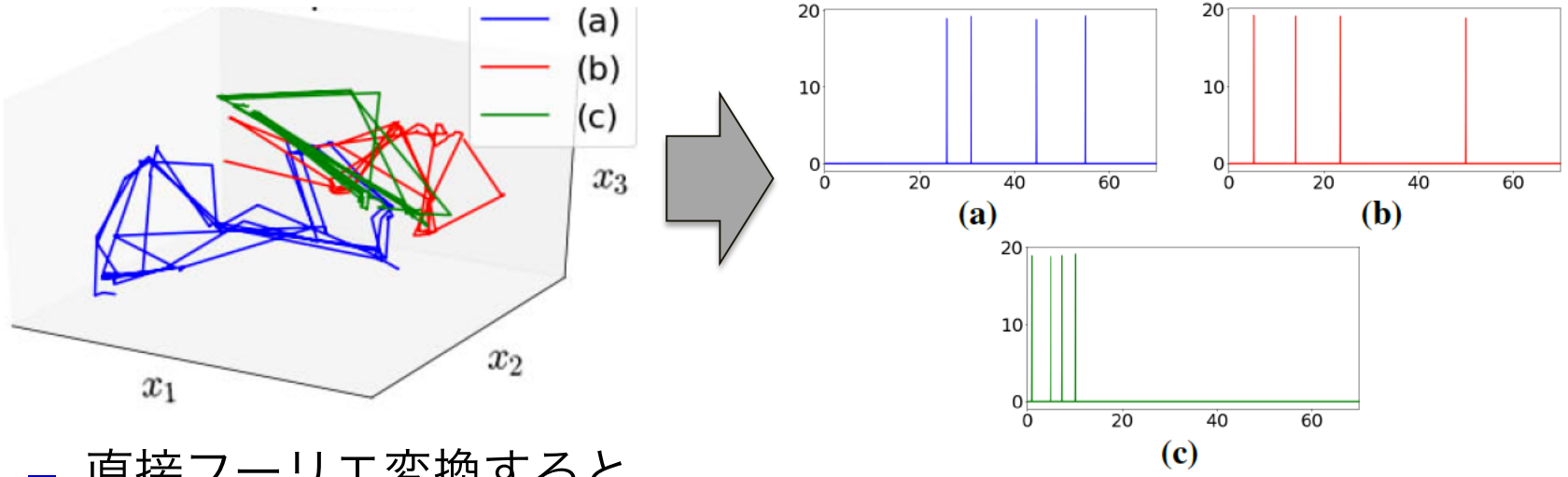
- 48次元の姿勢データを、GPLVMでP次元の潜在空間に非線形圧縮 (P=3)



- この潜在空間の時系列データの特徴をどう捉えるか?
  - 速い、遅い、微妙な震え、...
  - 「関数の特徴を捉える」問題

動画からの副詞の理解 (3)

- スペクトル混合カーネルの適用：カーネルの周波数空間での特徴を抽出

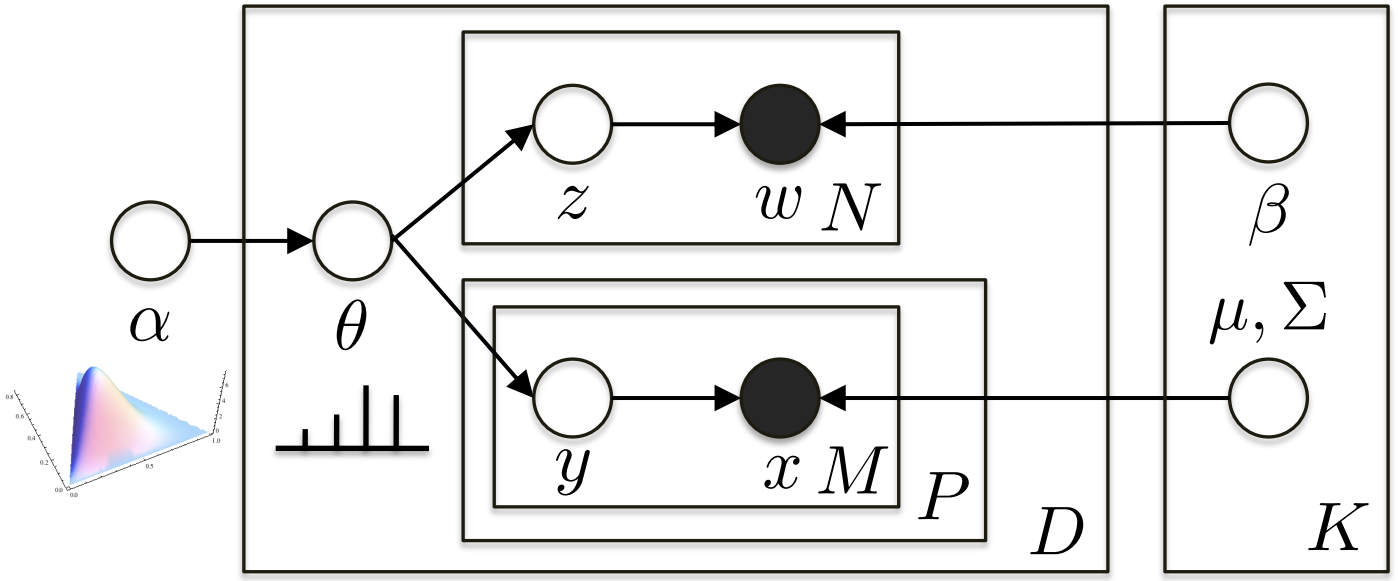


  - 直接フーリエ変換すると、
    関数の位相に依存＆FFTにはデータが不足

- 一方で、各動作には「しっかり」「堂々と」などの副詞が付与されている

Spectral Mixture LDA

- 抽出された動作の周波数空間での成分と、
  動画に付与された副詞の同時分布をモデル化
  → スペクトル混合潜在ディリクレ配分法 (SMLDA)



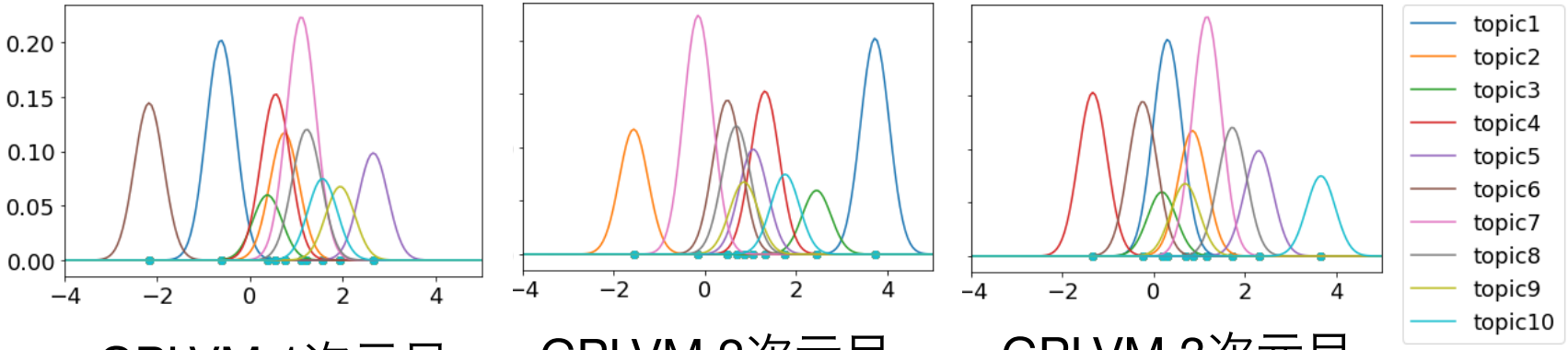
w：
副詞
x：
周波数
z, y：
トピック
θ：
潜在トピック分布

- 各トピックは、副詞の確率分布と周波数空間での
  ガウス分布の両方を持つ

Spectral Mixture LDA (2)

- 推定された副詞のトピックと、対応する対数周波数

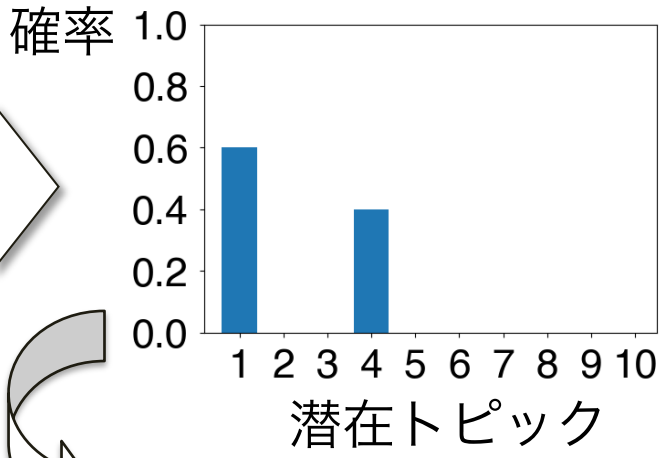
Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7
硬く	だらだら	大股で	痛そうに	楽しそうに	堂々と	ゆっくりと
奇妙に	元気なく	威嚇して	クタクタな	リズミカル	サッサッ	じっくりと
ふざけて	気だるく	どっしりと	足が痛い	調子よく	普通に	静々と
ぎこちなく	脱力して	忍んで	辛そうに	きびきび	力強く	確実に
不自然に	辛い	偉そうな	よろよろ	軽快に	颯爽と	一歩ずつ



GPLVM 1次元目　GPLVM 2次元目　GPLVM 3次元目

Spectral Mixture LDA (3)

- テストデータの動画からトピック分布と副詞を予測



動作の潜在トピック分布 θ

副詞	予測確率
ぎこちなく	0.056
痛そうに	0.054
不自然に	0.048
ゆっくり	0.036
ふざけて	0.035

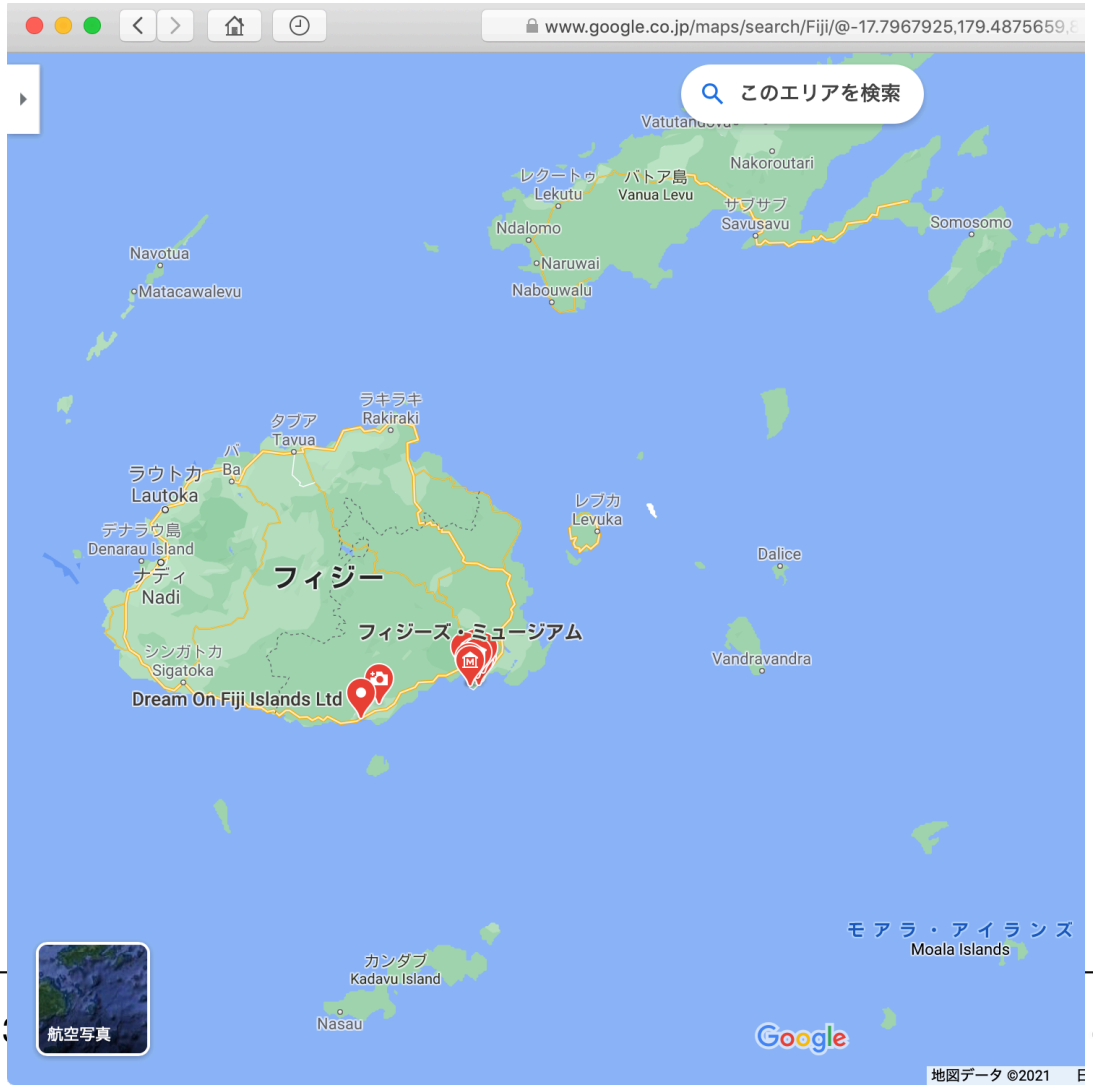
予測パープレキシティ：
149 (ユニグラム)
→ 32 (スペクトル混合LDA)

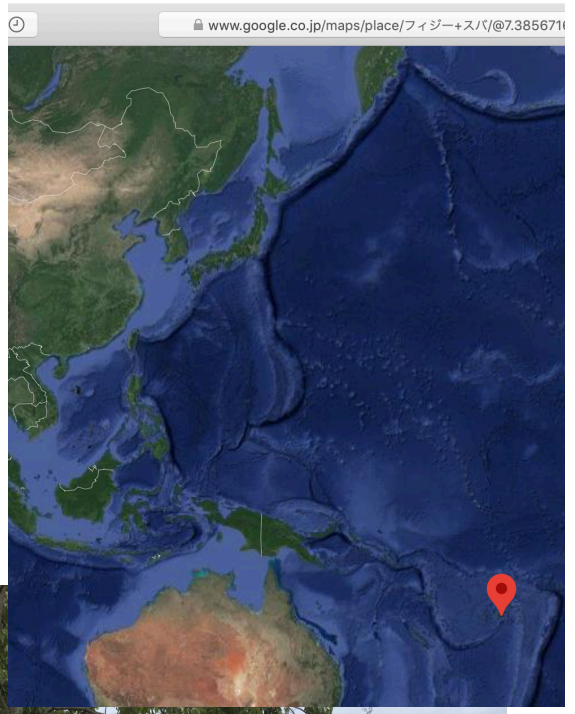
地理言語学

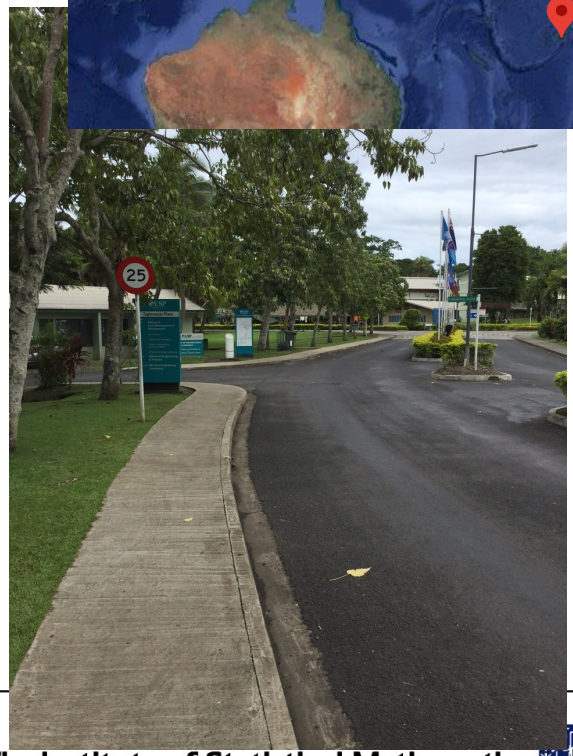
- 科研費国際共同研究強化(B)「 時空間を融合する：GISと数理モデルを用いた新たな言語変化へのアプローチ」(代表：菊澤律子 (民博))
  - NLPからは持橋、村脇さん(京大)が参加
- フィジーの方言のモデル化
  - Murawaki "Latent Geographical Factors for Analyzing the Evolution of Dialects in Contact” (EMNLP 2020)
  - 以下は、村脇さんの研究と若干違うモデル化

フィジー共和国

- 成田から直行便で9時間



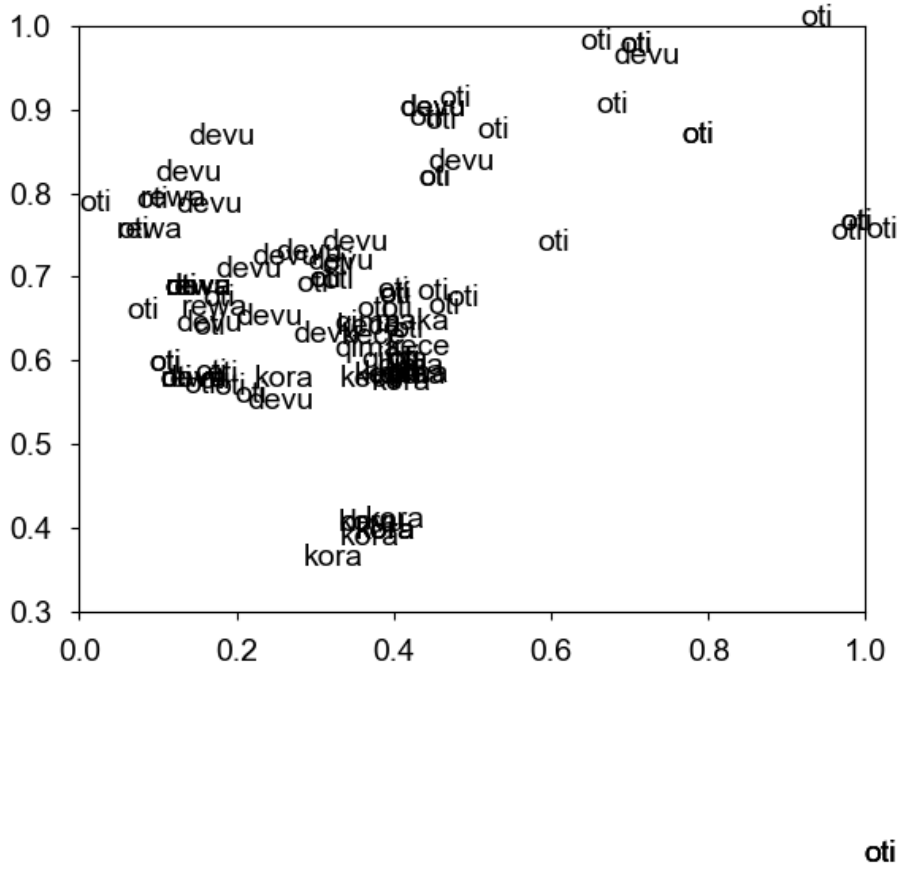




フィジー語方言の分布

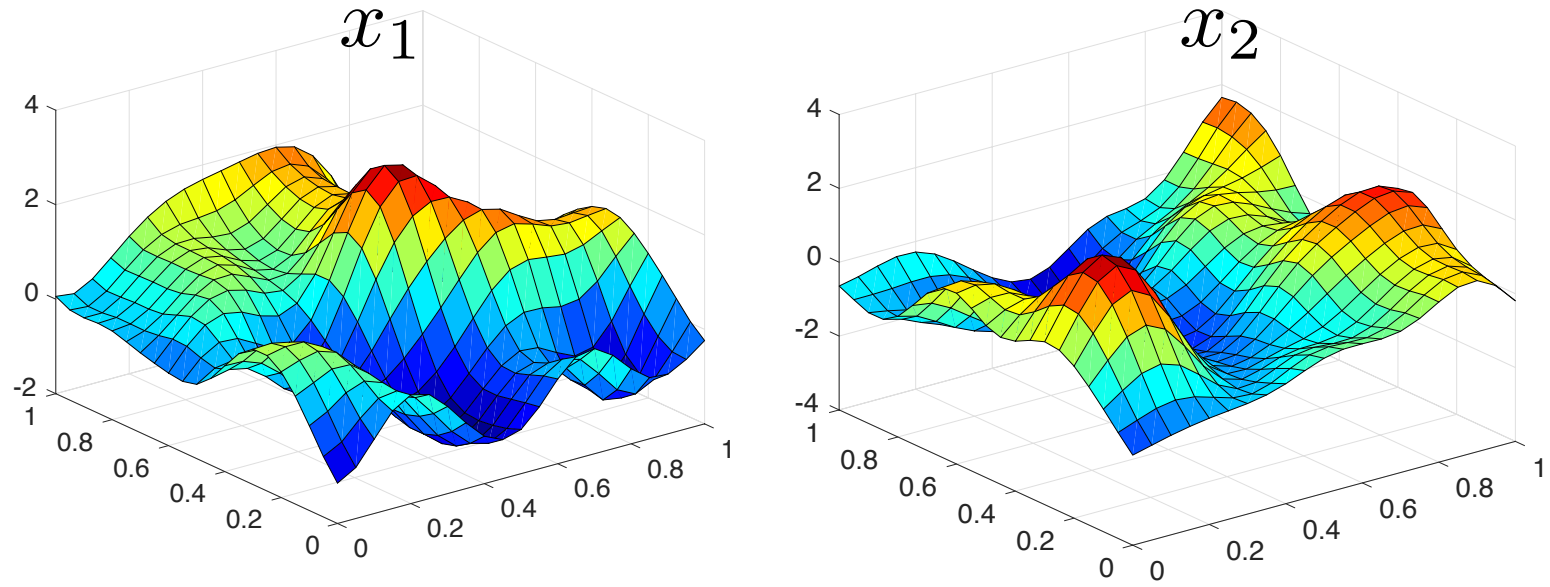
180.58	39.84	oti devu
182.76	39.67	oti rewa
183.34	39.17	oti
183.90	39.86	oti
184.13	39.87	rewa devu
184.61	38.85	oti devu
185.01	40.03	devu
185.27	38.75	oti devu rewa
185.54	39.32	oti rewa devu
186.13	39.09	devu
186.16	39.83	devu
186.46	39.19	rewa oti
186.66	38.71	oti devu
186.94	40.25	devu
187.21	39.07	oti
187.24	38.79	oti
187.42	38.74	oti

- どうやって方言の空間分布をモデル化する?



Variations in x_k

- x_k = (transformed) tendency of word k for a concept
- To model spatial variations of x_k, x_k is assumed to distribute according to a Gaussian process:



The Model

- $y_i = \{w_1, w_2, \cdots\}$: words used by a communalect *i* for a single concept
- $Y = (y_1, \cdots, y_N)$: collections of y_i over *N* different communalects

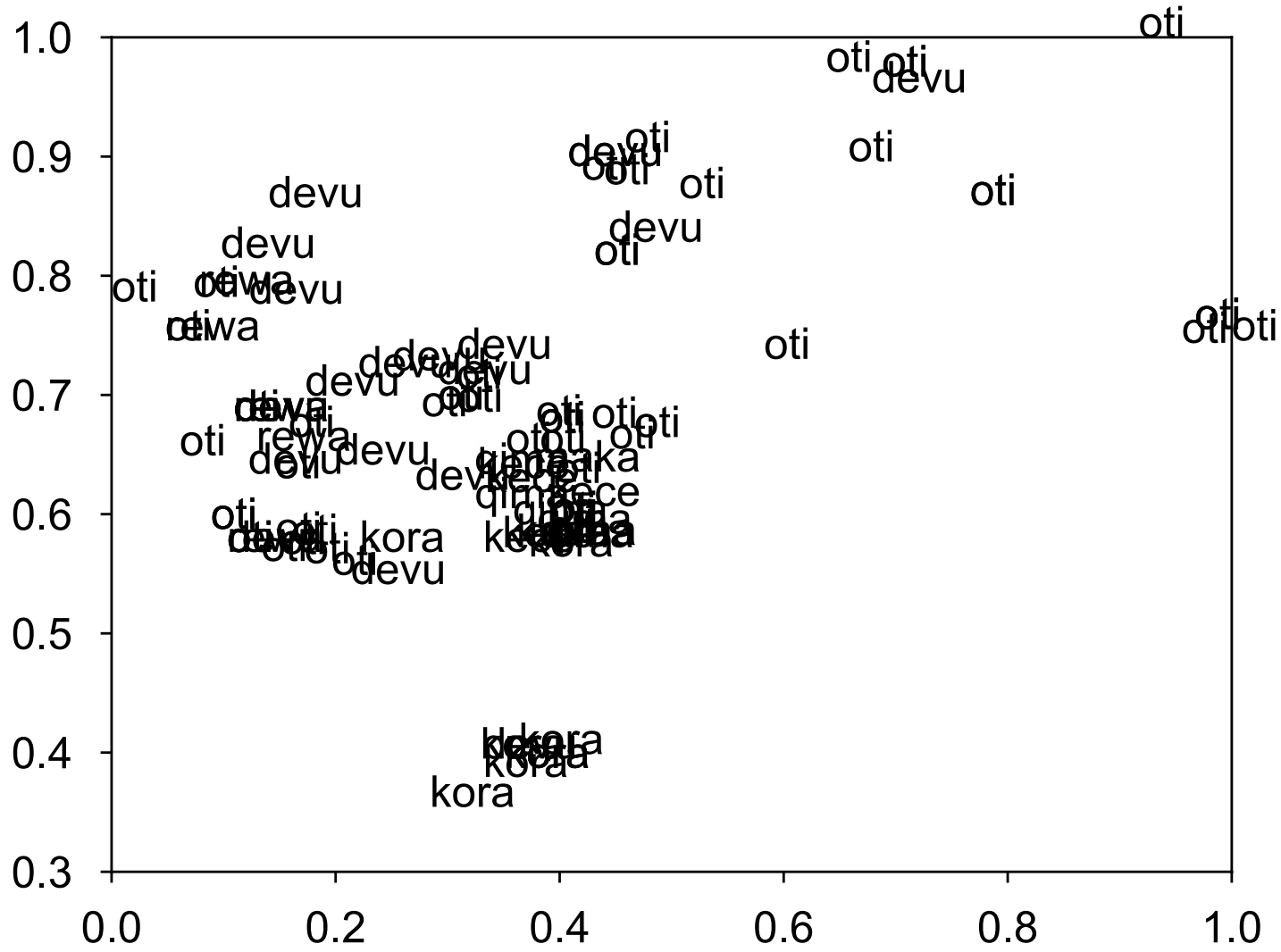
$$p(Y, X) = p(Y|X)p(X)$$

$$= \left(\prod_{i=1}^{N} \prod_{w \in y_i} p(w|\mathbf{x}_i) \right) \cdot \underbrace{\mathcal{N}(X|\mathbf{0}, \mathbf{K})}_{\text{Gaussian process}}$$

Softmax

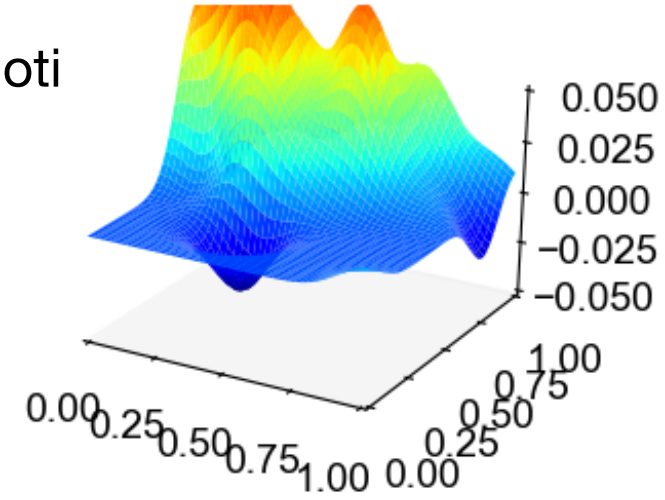
  - *X* can be sampled through an elliptical slice sampling (Murray+ 2010)

Examples on word 5

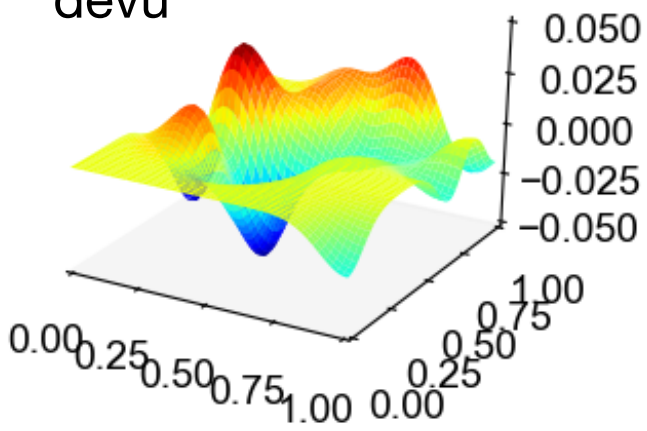


Examples on word 5

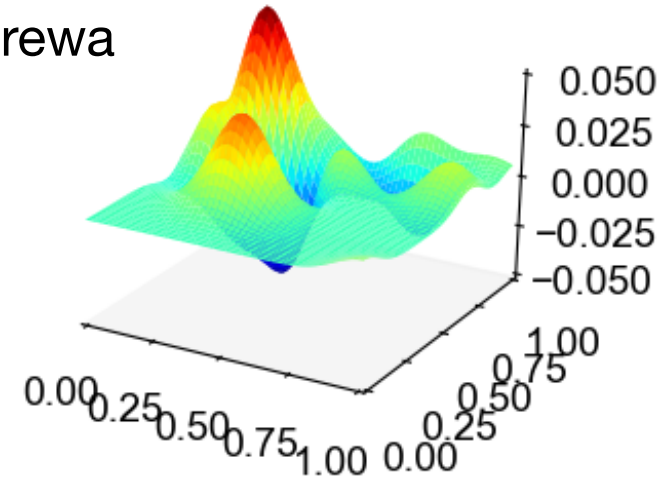
oti



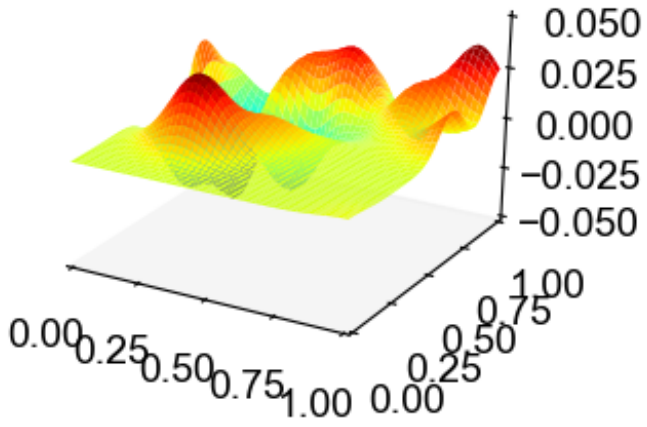
devu



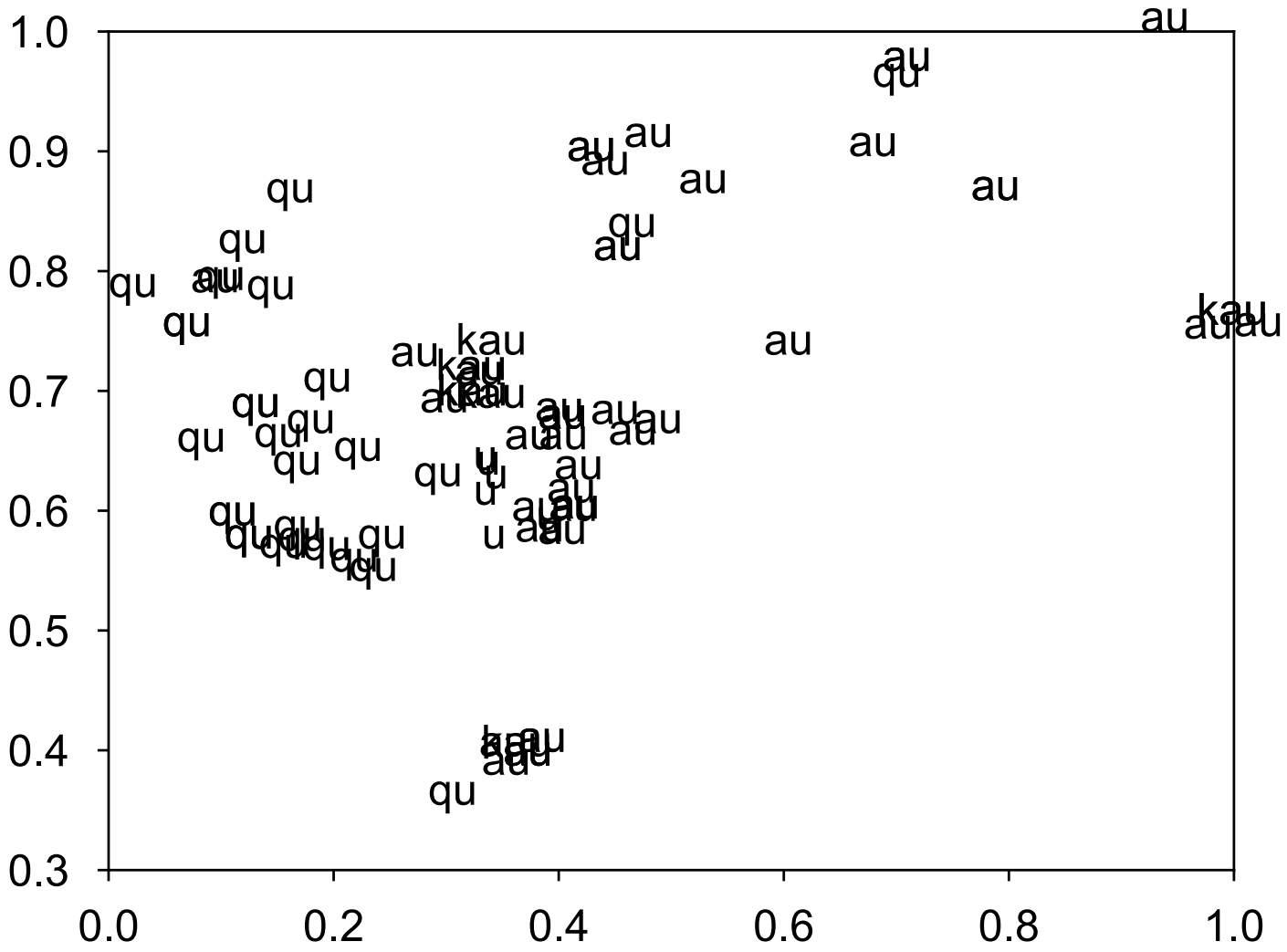
rewa



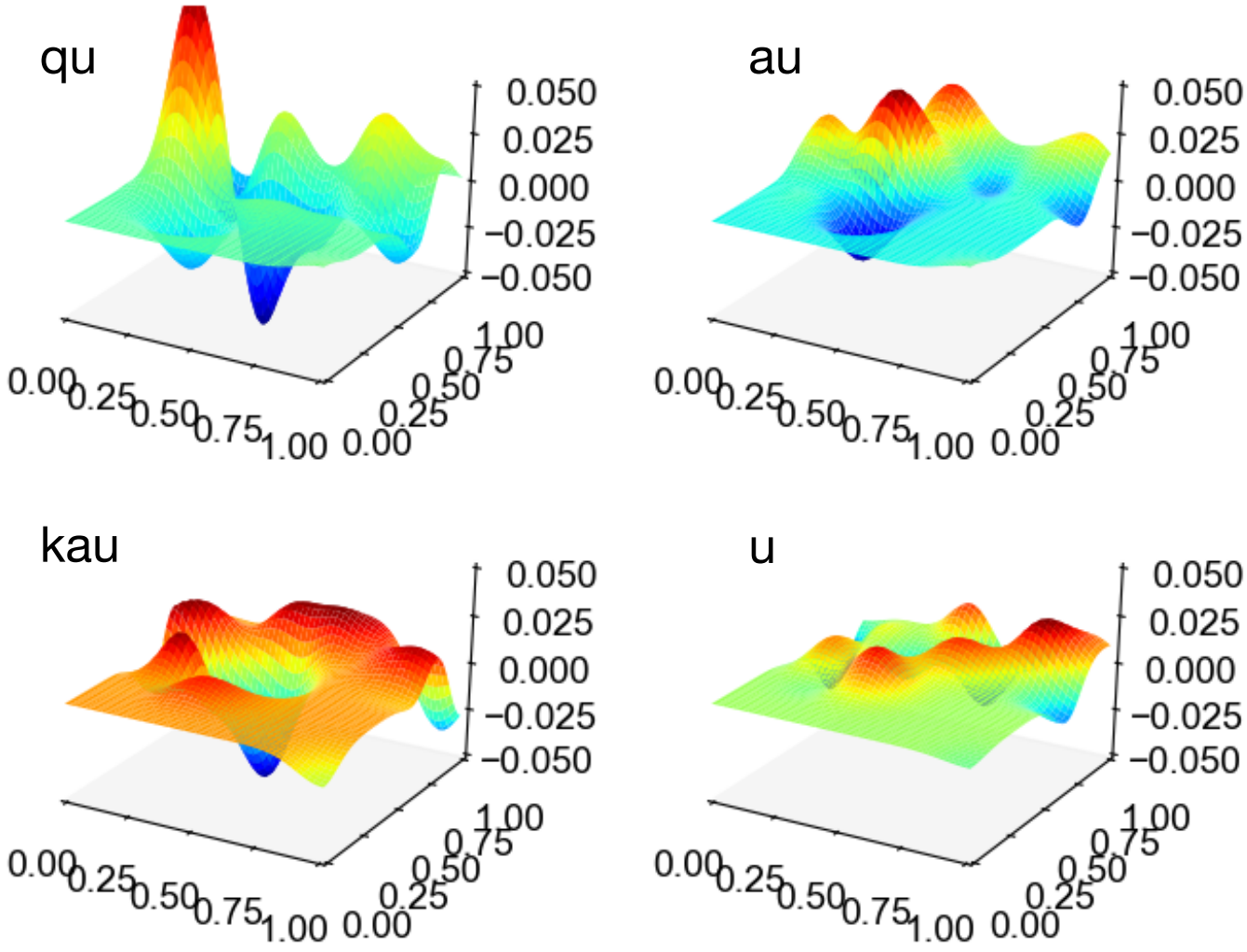
kora



Examples on word 2



Examples on word 2



まとめ

- ガウス過程：連続的に変化する関数を生成する確率過程
  - ガウス過程回帰＝カーネル法に基づくベイズ的な非線形回帰モデル
  - 空間データやロボティクスなど、自然言語処理でもこれから連続値を扱う必要
  - 教師なし学習も含め、さまざまな応用
- 深層学習はノード数→∞でガウス過程に一致
- 計算量はナイーブには $O(N^3)$ だが、様々な計算量削減法があり、実質 $O(N^2)$
- 詳しくは、『ガウス過程と機械学習』を参照のこと

終わり

- お疲れ様でした。
- 表紙が鹿なのは、著者の二人がどちらもNAIST出身のため.