

機械学習

2-2

工学院情報通信系

中原 啓貴

今日の内容

- 最小二乗学習
 - 線形モデル
 - 非線形モデル
 - 制約付き最小二乗学習
 - スパース学習
- + 上記内容の演習

クラス分類

- 識別的分類
 - 訓練データから学習することによる分類
- 生成的分類
 - データ生成確率を推定することによる分類
- 一般的には生成的分類のほうが難しい（パブニックの原理）
 - 先験的知識を利用できればその限りではない
 - 例：音声認識でいうと「人間のつくり」を利用
- モデル $q(x, y; \theta)$ におけるパラメータ θ を求める問題
 - 決定論的に扱う→頻度主義
 - 確率変数とみなす→ベイズ主義（後日扱う）

線形モデル

- 基底関数ベクトル $\phi(x)$ で表現

$$f_{\theta}(x) = \sum_{j=1}^b \theta_j \phi_j(x) = \boldsymbol{\theta}^T \boldsymbol{\phi}(x),$$

ここで, $\boldsymbol{\phi}(x) = (\phi_1(x), \dots, \phi_b(x))^T$ とし

$\boldsymbol{\theta} = (\theta_1, \dots, \theta_b)$ をパラメータベクトルとする

- 基底関数として多項式や三角多項式を用いて複雑な非線形関数を表現可能

- d次元の入力ベクトル $\boldsymbol{x} = (x^{(1)}, \dots, x^{(d)})^T$ に拡張

$$f_{\theta}(\boldsymbol{x}) = \sum_{j=1}^b \theta_j \phi_j(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}).$$

乗法モデル

- 1次元の既定関数の積で多次元の基底関数を構成

$$f_{\theta}(x) = \sum_{j_1=1}^{b'} \cdots \sum_{j_d=1}^{b'} \theta_{j_1, \dots, j_d} \phi_{j_1}(x^{(1)}) \cdots \phi_{j_d}(x^{(d)}),$$

ここで b' は各次元のパラメータ数.

- 全体のパラメータ数 $(b')^d$ 個 $\rightarrow$ 指数関数的に増加
「**次元の呪い**」という

加法モデル

- 1次元の基底関数の和で構成

$$f_{\theta}(x) = \sum_{k=1}^d \sum_{j=1}^{b'} \theta_{j,k} \phi_j(x^{(k)})$$

- 全体のパラメータ数 $b'd$ 個 → 線形的に増加
ただし複雑な関数の表現には不向き

カーネルモデル

- 線形モデル→訓練標本とは無関係に基底関数（多項式や三角関数）を用いる
- カーネルモデル→基底関数に入力標本を利用

$$f_{\theta}(x) = \sum_{j=1}^n \theta_j K(x, x_j),$$

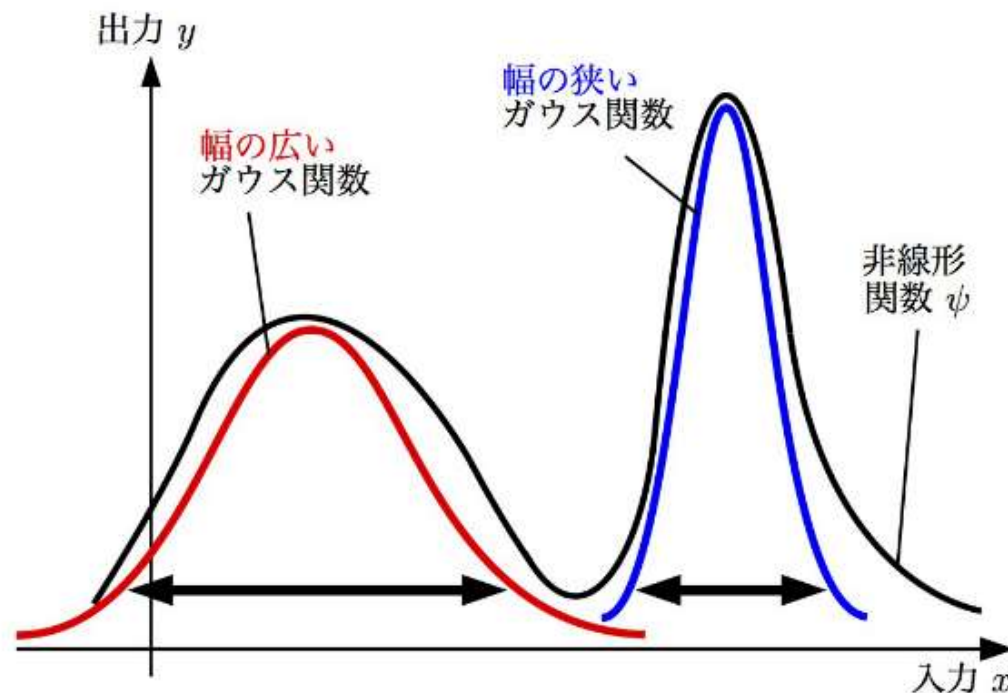
ここで $K(x, x_j)$ をカーネル関数といい, ガウスカーネル

$$K(x, c) = \exp\left(-\frac{\|x-c\|^2}{2h^2}\right),$$

がよく用いられる. ここで $\|\cdot\|$ は l_2 ノルム $\|x\| = \sqrt{x^T x}$, h はバンド幅, c は中心を表す.

例

- ガウスカーネルモデルは標本 x 毎にカーネルを配置し、その高さをパラメータとして学習
- 入力標本の近傍のみで近似するので次元の呪いを軽減できる



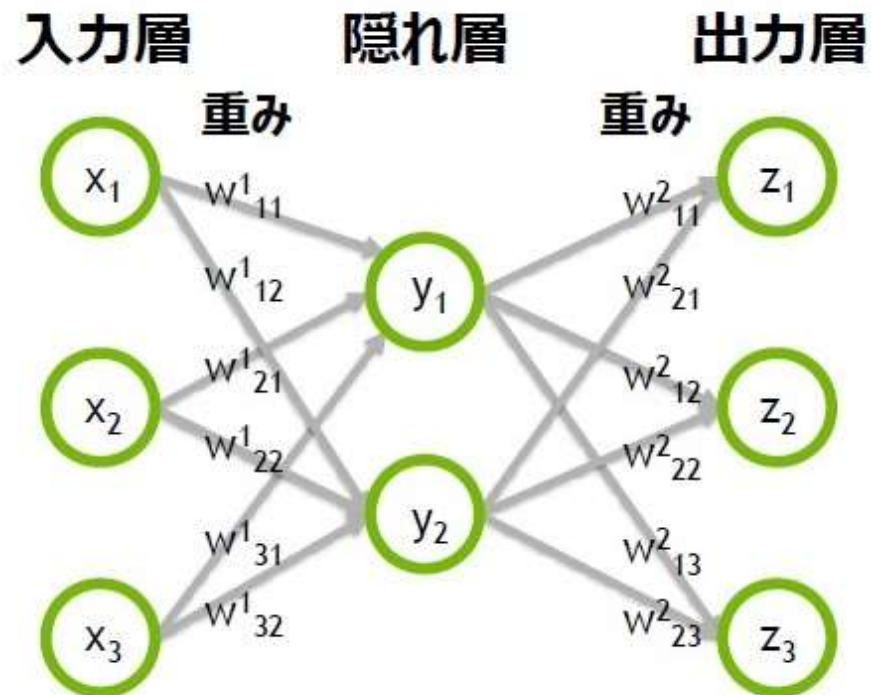
Source: <http://www.ykw.elec.keio.ac.jp/kakutyuu.html>

パラメトリックモデルと ノンパラメトリックモデル

- カーネルモデルは次元数 d に依存せず, 訓練標本 x のみで決まる
 - 入力集合からランダムに抽出した部分集合のみを用いて計算負荷を抑える
- パラメータベクトル $\theta = (\theta_1, \dots, \theta_b)$ に対して線形
→線形モデルの一種
- パラメータに対する線形モデル→パラメトリックモデル
- カーネルモデル→ノンパラメトリックモデル
- 入力 x は関数でもよい→カーネル法 (後曰述べる)

非線形モデル

- パラメータに関して非線形なモデル
 - 階層モデル、特にニューラルネットワークが有名
 - 活性化関数(シグモイド関数やtanh)を用いた階層モデル



最小二乗学習

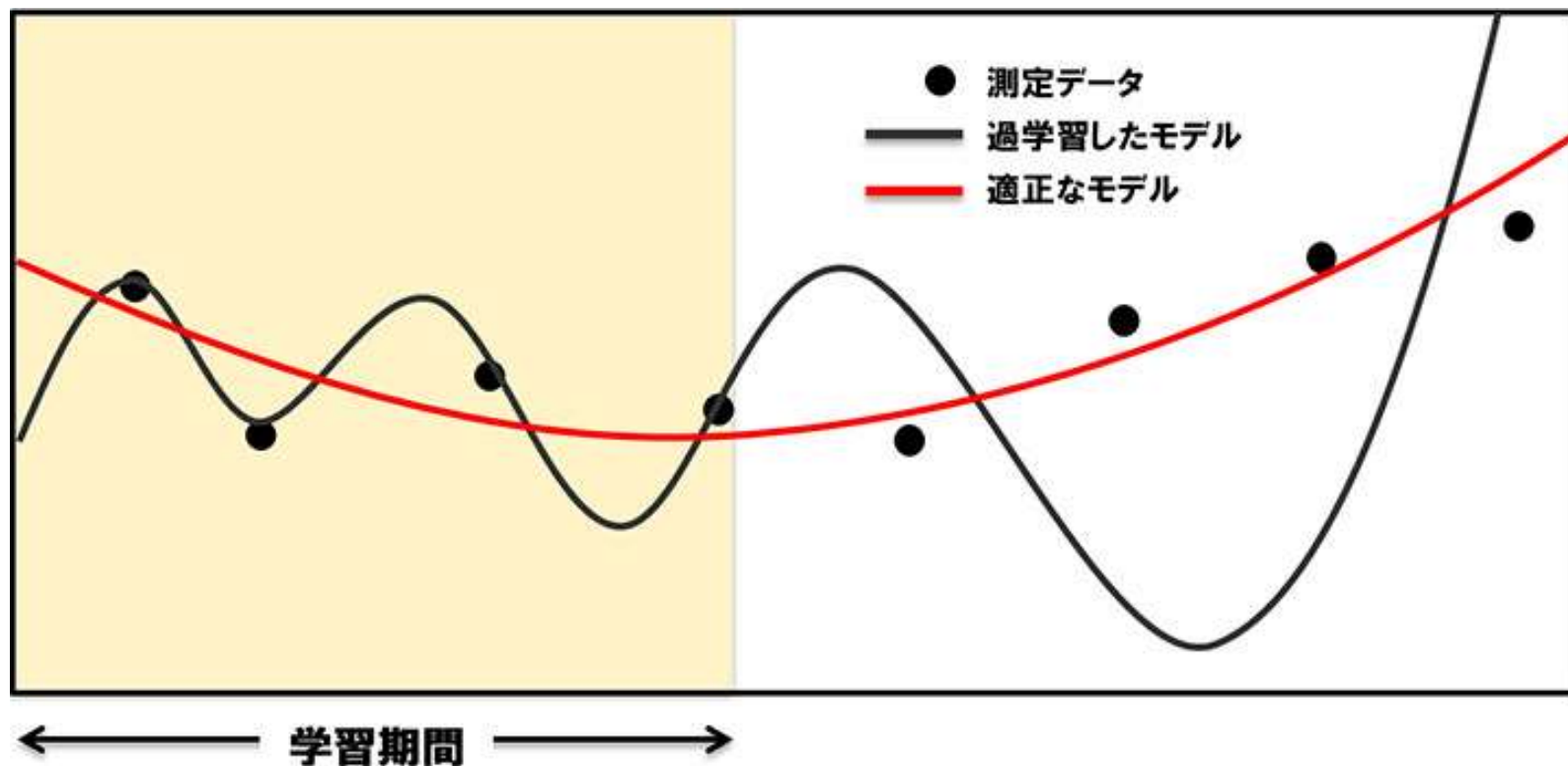
- (数式を用いて説明、必要であればメモ)

最小二乗解の性質

- 杉山将, "イラストで学ぶ機械学習", p.23-25.
- 中井悦司, "ITエンジニアのための機械学習入門", p.64-64.
- ポイント:
 - 2乗誤差は一般逆行列を用いて解析的に解ける
 - 厳密には条件（逆行列が存在するか）を確認
 - 係数（パラメータ）が訓練標本数を超える場合の議論

過学習(Overfitting)

- モデルが学習データに過剰に適合してしまい、評価データに対して性能が落ちること(c.f.汎化性能)



Source: http://www.nttdata.com/jp/ja/insights/trend_keyword/2014032701.html

l_2 制約付き学習(l_2 ノルム)

- パラメータ空間の原点を中心とする超球内で解を探索
→正則化項によるペナルティで過学習を防ぐ(**ロバストである**、ともいう)
- (数式を用いて説明、必要であればメモ)

l_1 制約付き学習(スパース学習)

- 代表アルゴリズム: LASSO(Least absolute shrinkage and selection operator)
 - Shrinkage → 次元圧縮(相関の無い説明変数の除去)
- l_2 制約付き学習と異なり解析的に解けない
 - 数値計算で求める必要がある

演習：最小二乗学習

- 線形モデル
- カーネルモデル
- L2制約を加えたカーネルモデル
- L1制約を加えた回帰（スパース学習）

https://github.com/HirokiNakahara/MachineLearning_Lecture

まとめ

- 分類法
 - 識別的分類、生成的分類
- モデルを定義
 - 線形モデル、非線形モデル
- 最小二乗学習
- L2制約付き学習
- L1制約付き学習(スパース学習)
- L1+L2制約付き学習 (**エラスティックネット回帰**)
 - 両方の性質を持つがハイパーパラメータが増える欠点も
- モデルの選択方法は？ 次回以降で解説