カーネル多変量解析

第2章　カーネル多変量解析の仕組み

２．１　カーネル関数とは何か：特徴抽出からの導入

２．２　正定値性からの導入

茨城大学工学部

佐々木稔

はじめに

- カーネル法
  - カーネル関数の重み付きの和で表したモデル
  - 過学習を抑えるため、正則化付きで最適化
- 第２章で知りたいこと
  - なぜカーネル関数が出てくるのか？
    - カーネル関数が満たすべき条件は？
  - 他にも複雑なカーネル関数が実現可能か？
    - サンプル点以外の点も使うことができないか？

線形モデル

- 基本の線形モデル
  - パラメータ（係数）**w** を決める
  - 直線的な関係しか表現できない

$$y = \mathbf{w}^T \mathbf{x}$$

- 1次の入力 x で、高次多項式を使うと

$$f(x) = \sum_{m=1}^{d} w_m x^m$$

  - 最適化したいパラメータは線形
    - 基本線形モデルと同じ最適化問題として解ける

特徴抽出

- 特徴抽出
  - 入力 x をある非線形変換で高次元に射影
  - 非線形関数 $\varphi(x) = (\varphi_1(x), \ldots, \varphi_d(x))^T$

$$f(\mathbf{x}) = \mathbf{w}^T \varphi(\mathbf{x}) = \sum_{m=1}^{d} w_m \varphi_m(\mathbf{x})$$

- 特徴抽出とカーネル多変量解析
  - 連続値でも離散値でも良い

入力 → 特徴抽出 → 特徴ベクトル → 構造抽出 → カーネル多変量解析

x

$\varphi(x)$

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k\left(\mathbf{x}^{(i)}, \mathbf{x}\right)$$

特徴抽出によるカーネル関数の定義

- 集合 X の要素 x, x' のカーネル関数 $k(x, x')$
  - x, x' の特徴ベクトルの内積として定義

$$k(\mathbf{x}, \mathbf{x}') = \varphi(\mathbf{x})^{\mathrm{T}} \varphi(\mathbf{x}') = \sum_{m=1}^{d} \varphi_m(\mathbf{x}) \varphi_m(\mathbf{x}')$$

  - 特徴抽出関数 $f(\mathbf{x}) = \mathbf{w}^T \varphi(\mathbf{x})$ はカーネル関数の和

$$f(\mathbf{x}) = \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) = \sum_i \alpha_i \varphi(\mathbf{x}_i)^{\mathrm{T}} \varphi(\mathbf{x})$$

  - となり、パラメータ w は以下の式で表される

$$\mathbf{w} = \left(\alpha_1 \varphi(\mathbf{x}_1), \alpha_2 \varphi(\mathbf{x}_2), \cdots, \alpha_n \varphi(\mathbf{x}_n)\right)$$

カーネル関数による線形モデルの表現

- 特徴抽出によるカーネル関数
  - 十分多くのデータによるカーネル関数の線形和
    - なければ過学習を引き起こす
  - サンプル点によるカーネル関数の線形和は？
    - ある特別な正則化を行うことで可能
  - リプリゼンター定理
    - 正則化を加えて最適化するための理論的な裏付け

リプリゼンター定理

- 損失（誤差）関数に正則化を加えて最適化する問題において、正則化項が $\lambda||w||^2$ の形をしていれば、最適解は $x^{(i)}$ をサンプル点として、

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k\left(\mathbf{x}^{(i)}, \mathbf{x}\right)$$

の形に書ける。

- 正則化項 $\lambda\alpha^{\mathrm{T}}K\alpha$ と $\lambda||w||^2$ は同じもの

$$\|w\|^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j \varphi\left(x^{(i)}\right)^T \varphi\left(x^{(j)}\right) = \alpha^T K \alpha$$

1変数多項式のカーネル関数

- 正則化項を付けて、二乗誤差を最適化

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i k\left(\mathbf{x}^{(i)}, \mathbf{x}\right) = \sum_{i=1}^{n} \alpha_i \sum_{m=1}^{d} \left(x^{(i)}\right)^m x^m$$

- サンプル数 n と多項式の次数 d の関係
  - $n > d$ のとき、d 次多項式を n 個の係数で表現
    - α_i 同士で係数を足し算する場合があり、冗長
  - $n < d$ のとき、最適化する係数が少ない
    - 特徴ベクトルの次元が大きいとパラメータ数が減少

類似度によるカーネル関数の定義

- 内積
  - 2つのベクトルの類似度を表現
- 類似度を表す関数
  - 正定値性をもてばカーネル関数として使用可能
- ある関数 $k(x, x')$ が正定値を持つとは
  - 任意の n 個の点 $x_1, \ldots, x_n$ からなるグラム行列 K
  - 任意の n 次元ベクトル $\alpha = (\alpha_1, \ldots, \alpha_n)$で、行列の二次形式が常に非負

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j K_{ij} \geq 0$$

$$K = \begin{pmatrix} k(x_1, x_1) & k(x_2, x_1) & \cdots & k(x_n, x_1) \\ k(x_1, x_2) & k(x_2, x_2) & \cdots & k(x_n, x_2) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_1, x_n) & k(x_2, x_n) & \cdots & k(x_n, x_n) \end{pmatrix}$$

カーネルと正定値性

- カーネル関数の計算
  - 特徴抽出、内積計算が必要

- 正定値性が分かっている関数では
  - 内積計算しなくても、関数の計算結果が分かる
  - 無限次元の特徴ベクトル間の内積も出せる
    - カーネルトリック

ガウスカーネル

- １次元ガウスカーネルの導出
  - $\boldsymbol{\varphi}$(x) は $\varphi_1(x)$, …, $\varphi_d(x)$ を持つ関数のベクトル

$$\varphi_z(x) = a \exp\left(-\beta'(z-x)^2\right)$$

  - この特徴ベクトルの内積は

$$k(x, x') = \int_{-\infty}^{\infty} \varphi_z(x)\varphi_z(x')dz$$

$$= a^2 \sqrt{\frac{\pi}{2\beta'}} \exp\left(-\frac{\beta'}{2}(x-x')^2\right)$$

  - ここで、$a = (2\beta'/\pi)^{1/4},\ \beta' = 2\beta$ とおくと、以下に一致

$$k(x, x') = \exp\left(-\beta\|x-x'\|^2\right)$$

ガウスカーネルの正定値性

- グラム行列の２次形式にカーネル関数を代入

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j \exp\left(-\beta\left(x_i - x_j\right)^2\right)$$

$$=\sum_{i=1}^{n}\sum_{j=1}^{n}\left\{\alpha_i \exp\left(-\beta x_i^2\right)\right\}\left\{\alpha_j \exp\left(-\beta x_j^2\right)\right\}\exp\left(2\beta x_i x_j\right)$$

$$=\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=0}^{\infty}\left\{\alpha_i \exp\left(-\beta x_i^2\right)\right\}\left\{\alpha_j \exp\left(-\beta x_j^2\right)\right\}\frac{\left(2\beta\right)^l x_i^l x_j^l}{l!}$$

$$=\sum_{l=0}^{\infty}\frac{\left(2\beta\right)^l}{l!}\left\{\sum_{i=1}^{n}\alpha_i \exp\left(-\beta x_i^2\right)x_i^l\right\}^2 \geq 0$$

多項式カーネル

- 多項式カーネル

$$k(\mathbf{x},\mathbf{x}') = \left(\mathbf{x}^T\mathbf{x}' + c\right)^p$$

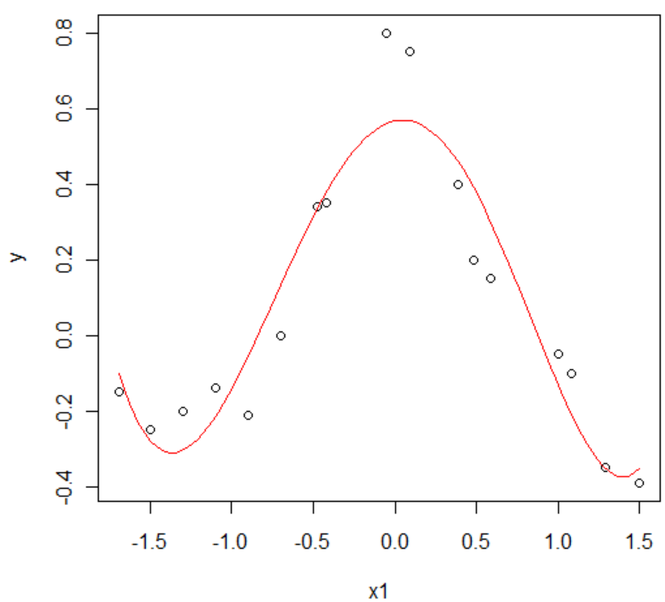
- 展開すると、内積計算は $O(d^p)$

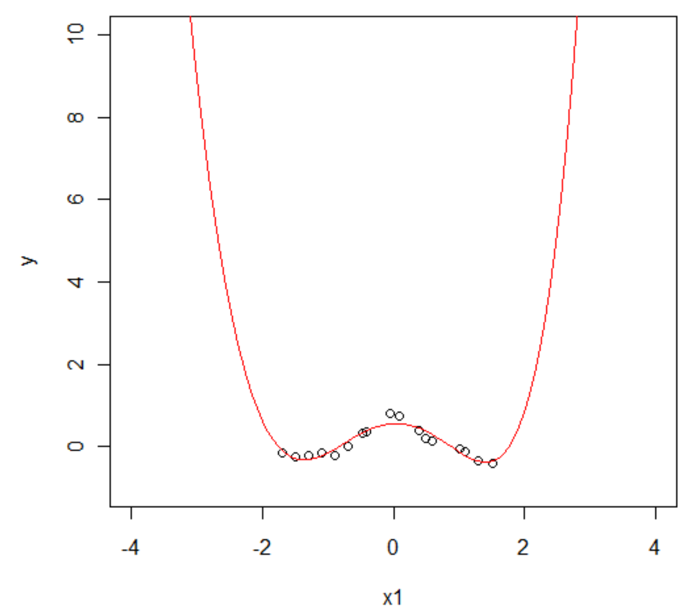
$$k(\mathbf{x},\mathbf{x}') = \sum_{m=0}^{p} \binom{p}{m} \left\{ \sum_{l=1}^{d} \left(x_l x_l' \right) \right\}^m c^{p-m}$$

  – 特徴ベクトルを計算し、内積計算
    - 膨大な計算量が必要

多項式カーネルによる関数推定

- 関数近似(λ＝0.01, c=1, *p*=5)のグラフ（左下）
  - 見た目はガウスカーネルに近い
- サンプル点の範囲外では
  - 急速に関数値が発散する（右下）





多項式カーネルによる関数推定を行うRプログラム

```
x1 <- c(-1.69, -1.5, -1.3, -1.1, -0.9, -0.7, -0.48, -0.42, -0.05, 0.09, 0.39,
0.48, 0.59, 1.0, 1.08, 1.29, 1.5)
y <- c(-0.15, -0.25, -0.2, -0.14, -0.21, 0, 0.34, 0.35, 0.8, 0.75, 0.4, 0.2,
0.15, -0.05, -0.1, -0.35, -0.39)
myPolynomialKernelMatrix <- function(x, degree=5, scale=1, offset=1)
(((scale) * (x %*% t(x) + offset))^degree)
mypoly <- function(x1, x2, degree=5, scale=1, offset=1) (((scale) * (x1 *
x2) + offset)^degree)

k <- myPolynomialKernelMatrix(x1)
lambda <- 0.01
k2 <- k + lambda * diag(17)
alpha <- solve(k2) %*% y
f1 <- function(x) sapply(x, function(y) mypoly(x1, y) %*% alpha)
plot(x1,y)
curve(f1,col="red",add=T)
```

ガウスカーネルによる関数推定

- ガウスカーネルを使った関数近似

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k\left(\mathbf{x}^{(i)}, \mathbf{x}\right)$$

  – サンプル領域の外側まで広げると
    - *f*(*x*) は 0 に近づく