

カーネル多変量解析

第1章

現代の多変量解析とは

新納浩幸

現代の多変量解析の特徴

(1) データマイニング

大量のデータの中にある隠れた構造を取り出す

高次元、実数値だけでない多様なデータ



カーネル法

(2) テーラーメイドな多変量解析

データ解析の目的は多様



目的に応じた解析手法の提案を手助け
(本書の目的の1つ)

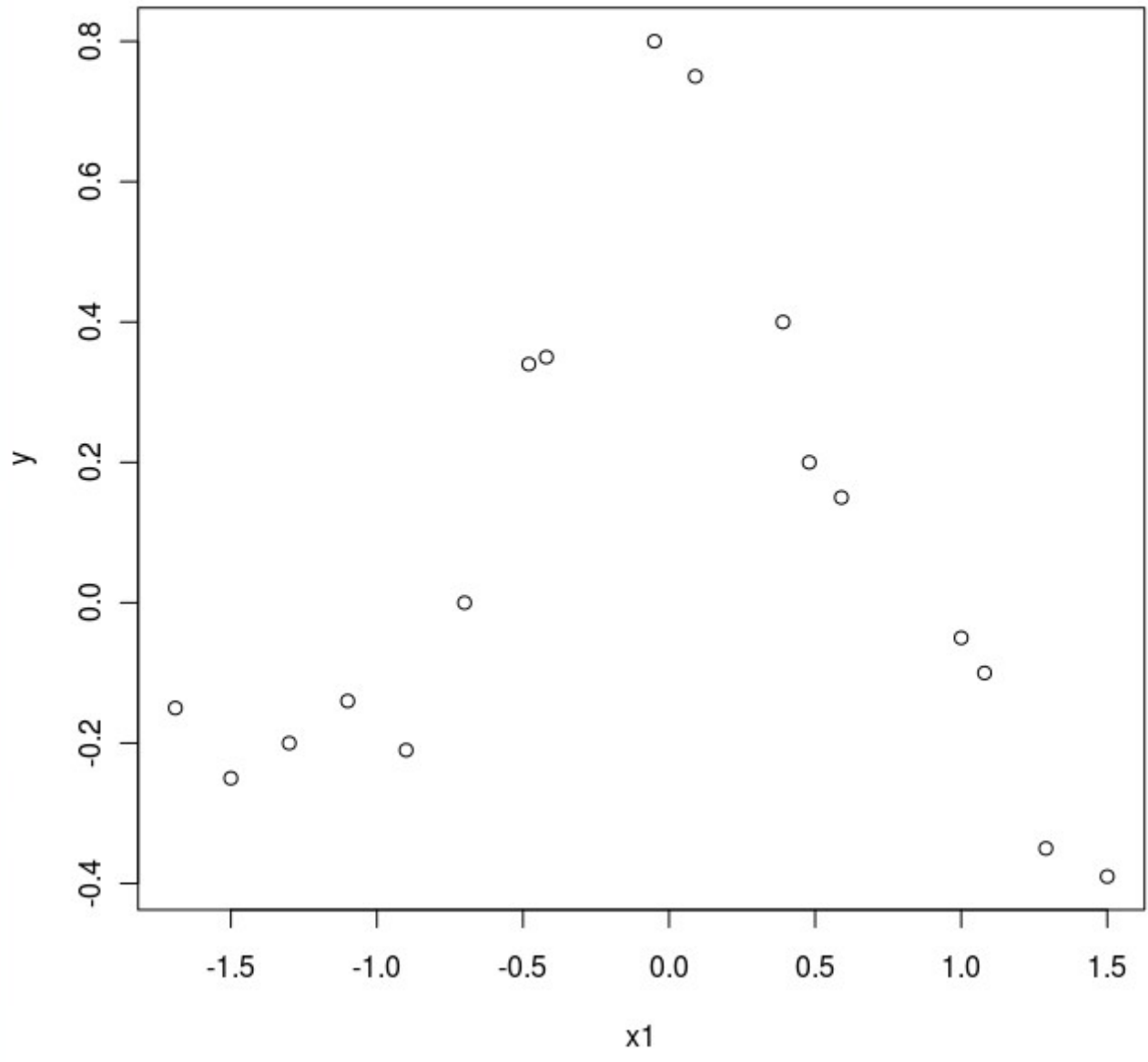
関数推定

$$f : R^d \rightarrow R$$

教師データ(観測データ) は n 個

$$\{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})\}$$

R Graphics: Device 2 (ACTIVE)



R × E(1)

```
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)
```

```
plot(x1,y)
```

線形モデル(1)

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

訓練データから w を推定する

 最小二乗法

$$R(\mathbf{w}) = \sum_{j=1}^n (y^{(j)} - \mathbf{w}^T \mathbf{x}^{(j)})^2$$

最小にする w が求めるもの

線形モデル(2)

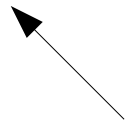
$$\mathbf{y} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \cdot \\ \cdot \\ \cdot \\ y^{(n)} \end{pmatrix} \quad X = \begin{pmatrix} \mathbf{x}^{(1)T} \\ \mathbf{x}^{(2)T} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{x}^{(n)T} \end{pmatrix} \quad n \text{ 行 } d \text{ 列 の行列}$$

$$R(\mathbf{w}) = (\mathbf{y} - X\mathbf{w})^T (\mathbf{y} - X\mathbf{w})$$

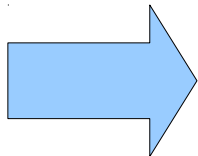
\mathbf{w} は d 次元の縦ベクトル

線形モデル(3)

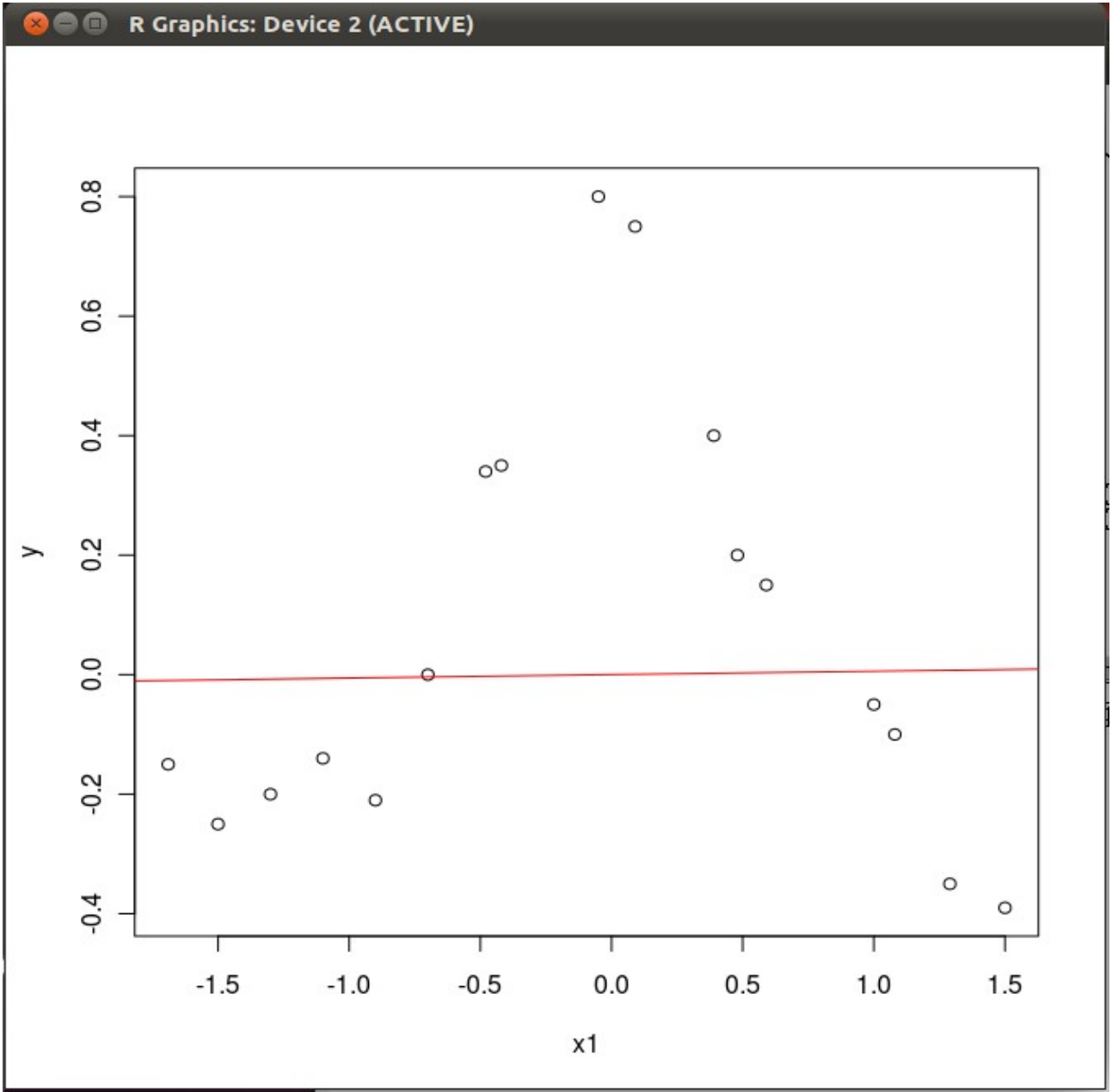
$$\frac{\partial R}{\partial \mathbf{w}} = 2X^T (\mathbf{y} - X\mathbf{w}) = \mathbf{0}$$



ベクトルで微分(別スライドで解説)



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



R × (2)

```
m0 <- lm(y ~ -1 + x1)
```

```
abline(m0,col="red")
```

カーネル法

カーネル関数 $k(\mathbf{x}, \mathbf{x}')$

入力 \mathbf{x} と 与えられている点 \mathbf{x}' との近さを表す

(例) $k(\mathbf{x}, \mathbf{x}') = \exp(-\beta \|\mathbf{x} - \mathbf{x}'\|^2)$

カーネル法によるモデル化

$$y = \sum_{j=1}^n \alpha_j k(\mathbf{x}^{(j)}, \mathbf{x})$$

パラメータは α

カーネル法の残差(1)


$$\begin{aligned} R(\alpha) &= \sum_{i=1}^n \left(y^{(i)} - \sum_{j=1}^n \alpha_j k(\mathbf{x}^{(j)}, \mathbf{x}_i) \right)^2 \\ &= (\mathbf{y} - K\alpha)^T (\mathbf{y} - K\alpha) \end{aligned}$$

$$K = \begin{pmatrix} k(\mathbf{x}^{(1)}, \mathbf{x}^{(1)}) & k(\mathbf{x}^{(2)}, \mathbf{x}^{(1)}) & \dots & k(\mathbf{x}^{(n)}, \mathbf{x}^{(1)}) \\ k(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) & k(\mathbf{x}^{(2)}, \mathbf{x}^{(2)}) & \dots & k(\mathbf{x}^{(n)}, \mathbf{x}^{(2)}) \\ \dots & \dots & \dots & \dots \\ k(\mathbf{x}^{(1)}, \mathbf{x}^{(n)}) & k(\mathbf{x}^{(2)}, \mathbf{x}^{(n)}) & \dots & k(\mathbf{x}^{(n)}, \mathbf{x}^{(n)}) \end{pmatrix}$$

カーネル法の残差(2)

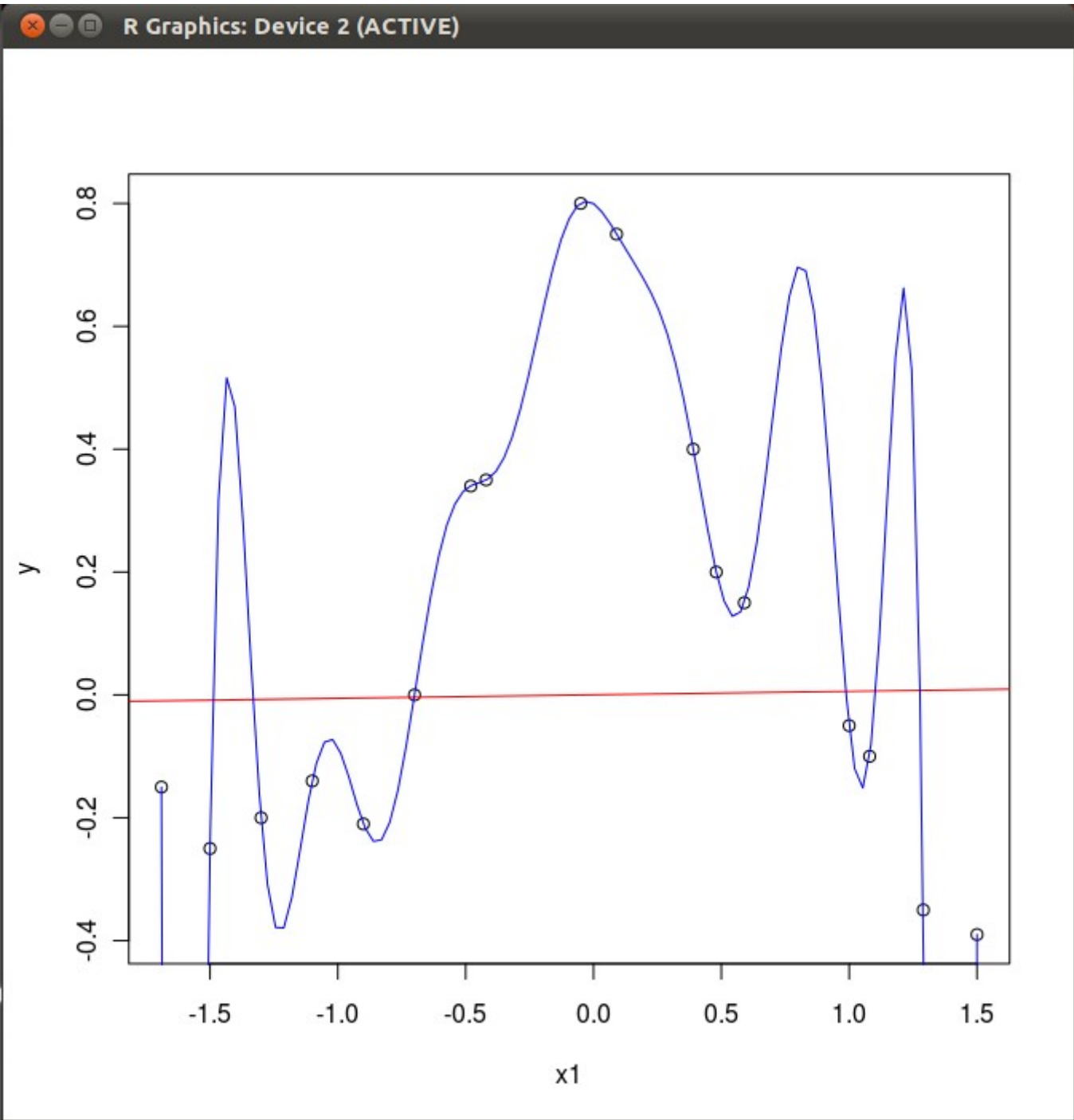
$$R(\alpha) = (\mathbf{y} - K\alpha)^T (\mathbf{y} - K\alpha)$$

$$\frac{\partial R}{\partial \alpha} = -2K^T (\mathbf{y} - K\alpha) = \mathbf{0}$$

 $\alpha = (K^T K)^{-1} K^T \mathbf{y}$

K は対象行列なので $K^T = K$

$$\alpha = K^{-1} \mathbf{y}$$



R メモ(3)

```
rbf <- rbfdot(sigma=1)
k <- kernelMatrix(rbf, x1)
alpha <- solve(k) %*% y
gauss <- function(x1,x2) exp(-1.0 * (x1-x2)^2)
f1 <- function(x) sapply(x, function(y) gauss(x1, y) %*% alpha)
curve(f1, col="blue", add=TRUE)
```

パッケージ kernlab をインストールする

```
install.packages("kernlab")
```

パッケージ kernlab を読む込む

```
library("kernlab")
```

正則化(1)

カーネル法の汎化能力を少し落として、より適切に関数近似を行うために、残差に**正則化項**を加える

$$R(\alpha) = (\mathbf{y} - K\alpha)^T (\mathbf{y} - K\alpha) + \underbrace{2\lambda\alpha^T K\alpha}$$

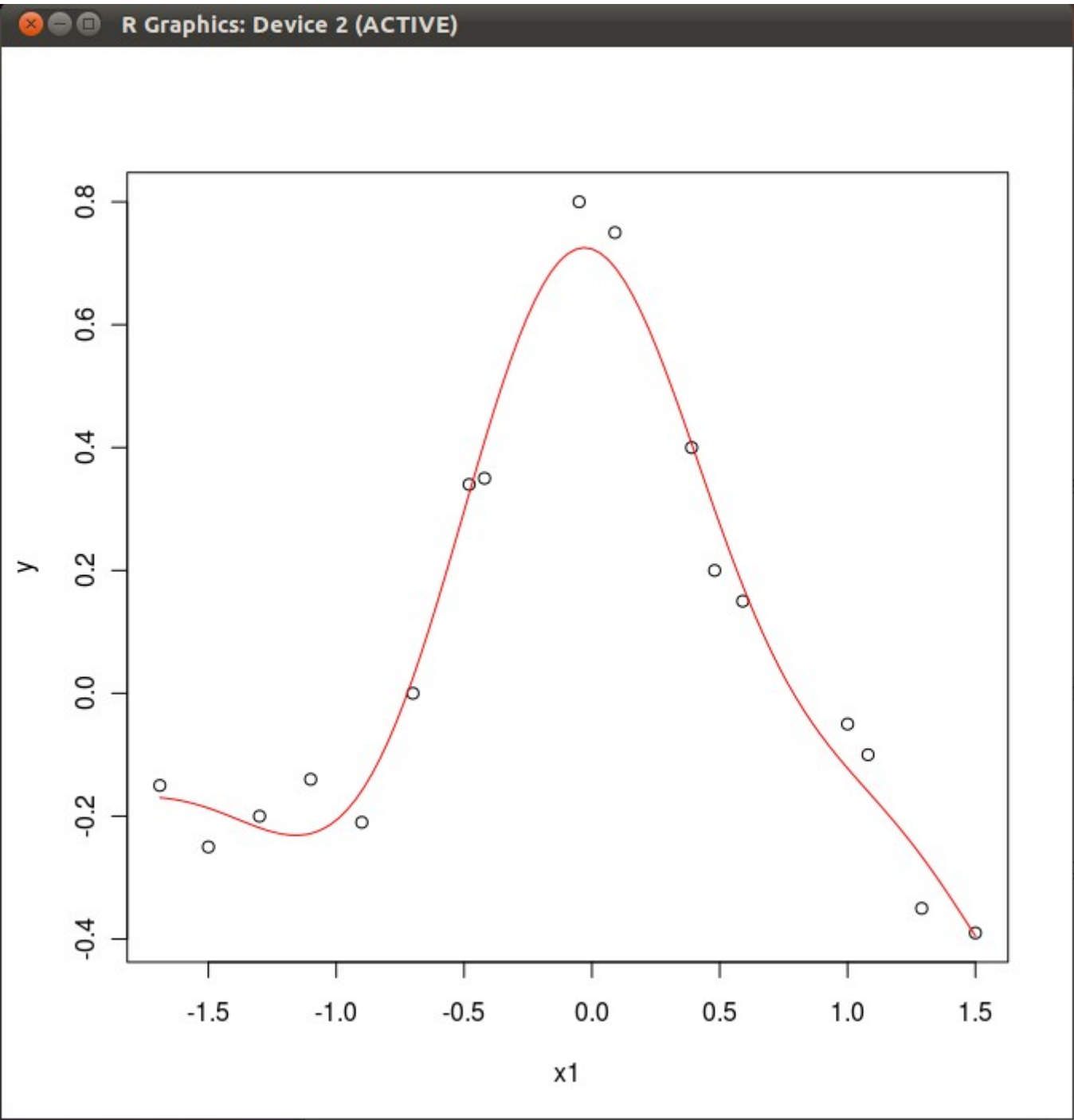
λ : 正則化パラメータ

正則化(2)

$$\frac{\partial R}{\partial \alpha} = -2K^T(\mathbf{y} - K\alpha) + 2\lambda K\alpha = \mathbf{0}$$



$$\alpha = (K + \lambda I)^{-1} \mathbf{y}$$



R×ε(4)

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

カーネル法の利点

(1) サンプル数に応じてモデルを複雑にできる

パラメータ α の次元はサンプル数

(2) 線形性と非線形性の両面をもつ

データに対しては非線形、パラメータに対しては線形

(3) カーネル関数の利用

非数値データ、高次元データもカーネル関数の値さえ
求めればカーネル法は利用可能

(4) カーネル関数のモジュール化

カーネル関数の計算とその後の処理が分離

カーネル法の応用分野

パターン認識のタスク

文字認識、音声認識 など

Web 上のサービス

大量テキスト、自然言語処理、XML

バイオインフォマティクス

DNA やアミノ酸配列からの情報抽出
有用な遺伝子の発見

カーネル法の種類

