データ解析
第十一回「ノンパラメトリック回帰」

鈴木　大慈
理学部情報科学科
西八号館 W707 号室
s-taiji@is.titech.ac.jp

休講情報

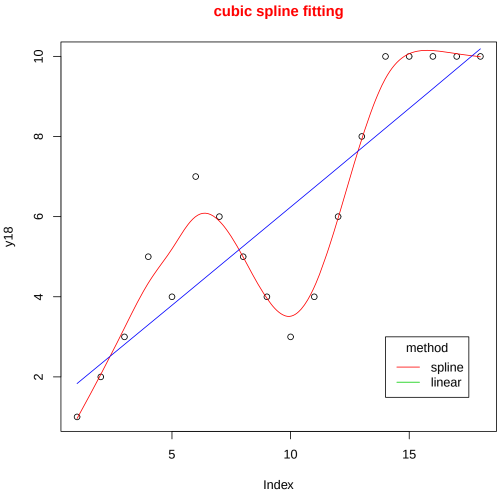
7/7 は休講

今日の講義内容

ノンパラメトリック回帰手法

- カーネル回帰 (Nadaraya-Watson 推定量)
- k-近傍回帰
- スプライン回帰

ノンパラメトリック回帰



ノンパラメトリック回帰の問題設定

$$Y_i = f(X_i) + \epsilon_i$$

ただし $\{\epsilon_i\}_{i=1}^n$ は雑音で $\mathrm{E}[\epsilon_i] = 0$ かつ $\sigma^2 = \mathrm{E}[\epsilon_i^2]$ で i.i.d..

- (X_i, Y_i) $(i = 1, \ldots, n)$ から $f(x)$ を推定したい.
- $\mathrm{E}[\epsilon] = 0$ なので, $f(x)$ は

$$f(x) = \mathrm{E}[Y|x]$$

  である.
- f は滑らかさぐらいしか仮定しない.

構成

1. Nadaraya-Watson 推定量
2. k-近傍回帰法
3. B-スプライン

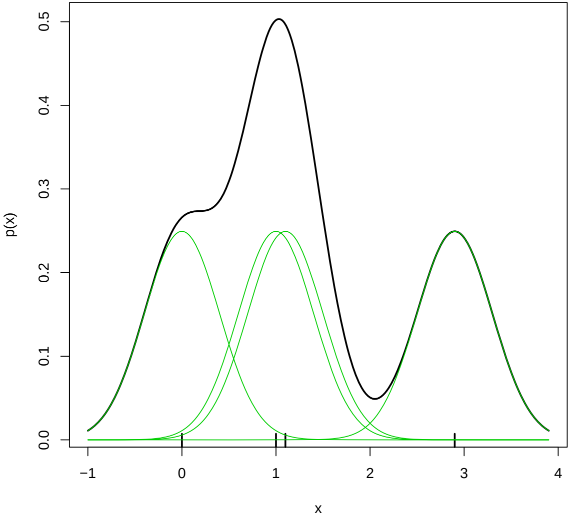
Nadaraya-Watson 推定量

カーネル密度推定の回帰版.

$$\hat{f}(x) = \frac{\sum_{i=1}^{n} Y_i K\left(\frac{X_i - x}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)}$$

1. $h > 0$ はバンド幅と呼ぶ．適切に選択する必要がある.
2. K は次の性質を満たすものとする:

$$\int K(x)\mathrm{d}x = 1, \quad \int xK(x)\mathrm{d}x = 0, \quad \int x^2 K(x) > 0.$$



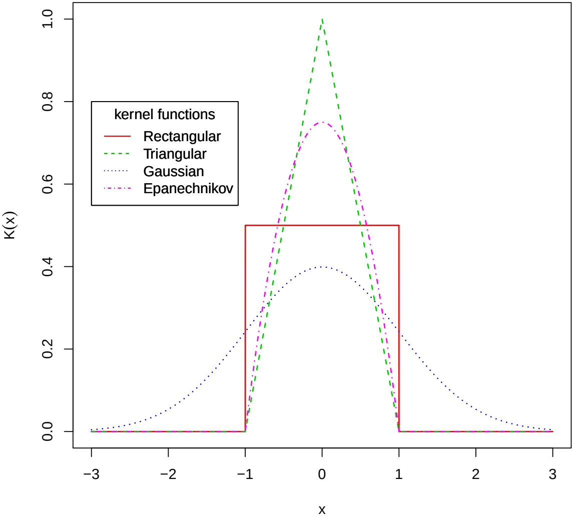
p(x)
0.0
0.1
0.2
0.3
0.4
0.5
−1
0
1
2
3
4
x

カーネルの種類

カーネル密度推定量と同様に次のようなカーネル関数がよく用いられる.

1. Gaussian:
$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right).$$
2. Rectangular:
$$K(x) = \begin{cases} \frac{1}{2} & (|x| \leq 1), \\ 0 & (\text{otherwise}). \end{cases}$$
3. Triangular:
$$K(x) = \begin{cases} |x| & (|x| \leq 1), \\ 0 & (\text{otherwise}). \end{cases}$$
4. Epanechnikov:
$$K(x) = \begin{cases} \frac{3}{4}(1 - x^2) & (|x| \leq 1), \\ 0 & (\text{otherwise}). \end{cases}$$

カーネルの種類


kernel functions
Rectangular
Triangular
Gaussian
Epanechnikov
K(x)
x
0.0
0.2
0.4
0.6
0.8
1.0
−3
−2
−1
0
1
2
3

Nadaraya-Watson 推定量の簡単な導出

説明変数 x を固定したもとでの Y の期待値を求めたい:

$$\mathrm{E}[Y|x] = \int yp(y|x)\mathrm{d}y = \frac{\int yp(y,x)\mathrm{d}y}{p(x)}.$$

ここで，$p(x)$, $p(y,x)$ をカーネル密度推定しよう:

$$p(y,x) \simeq \frac{1}{nh^2}\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) K\left(\frac{Y_i - y}{h}\right),$$

$$p(x) \simeq \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right).$$

これを用いると，

$$\int yp(y,x)\mathrm{d}y \simeq \frac{1}{nh^2}\int y\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) K\left(\frac{Y_i - y}{h}\right)\mathrm{d}y$$

$$= \frac{1}{nh}\sum_{i=1}^{n} Y_i K\left(\frac{X_i - x}{h}\right)$$

である．なお，$\int yK(y)\mathrm{d}y = 0$ の条件を用いた．あとは，これらを条件付き期待値の式に代入して Nadaraya-Watson 推定量を得る．

Nadaraya-Watson 推定量の漸近的性質

漸近分布

Theorem

σ^2 を $Var[\epsilon]$ とする．

$$\sqrt{nh}\Big(\hat{f}(x) - f(x) - h^2 \int u^2 K(u)\mathrm{d}u B(x)\Big) \xrightarrow{d} N\left(0, \frac{\sigma^2 \int K^2(u)\mathrm{d}u}{p(x)}\right),$$

ただし，

$$B(x) = \frac{1}{2} f''(x) + \frac{f'(x)p'(x)}{p(x)}.$$

特に $n \to \infty$ で $h \to 0$ なら，右辺は $\sqrt{nh}\Big(\hat{f}(x) - f(x)\Big)$ とできる．

Nadaraya-Watson 推定量の漸近的性質

平均二乗誤差 $\mathrm{MSE}(\hat{p}(x), h) := \mathrm{E}[(\hat{f}(x) - f(x))^2]$.

Theorem

$$\mathrm{MSE}(\hat{p}(x), h) \to h^4 \int u^2 K(u)\mathrm{d}u B^2(x) + \frac{\int K^2(u)\mathrm{d}u\sigma^2}{nhp(x)}.$$

これより，漸近的に最適な h は

$$h^* = \left(\frac{1}{n}\frac{\int K^2(u)\mathrm{d}u\sigma^2}{4\int u^2 K(u)\mathrm{d}u B^2(x)p(x)}\right)^{1/5}$$

であり，$h^* = O(n^{-1/5})$ で $\mathrm{MSE}(\hat{p}(x), h) = O(n^{-4/5})$ である．これはカーネル密度推定と同じ収束レートである．

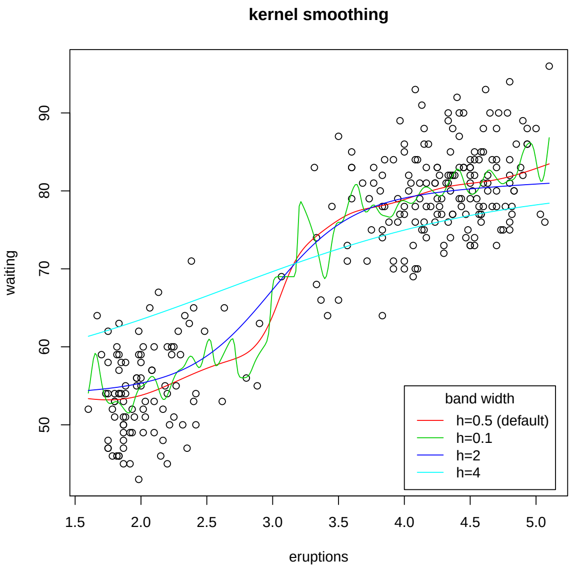
多変量への拡張

H: 正定値対称行列.
$\|x\|_H^2 := x^\top Hx$ とする.

多変量の拡張は以下のようにすれば良い:

$$\hat{f}(x) = \frac{\sum_{i=1}^n Y_i K(\|X_i - x\|_H)}{\sum_{i=1}^n K(\|X_i - x\|_H)}.$$

Nadaraya-Watson 推定量の実験



R のコード

```
ks.reg <- ksmooth(x, y, kernel = c("box", "normal"),
                  bandwidth = 0.5, x.points)
```

- kernel: rectangular (box) と Gaussian (normal) で選択可.
- x.points: テスト点の設定. 指定しなければ観測データ点における予測値を返す.

ks.reg には値を予測した点の x と y が格納されている.

構成

1. Nadaraya-Watson 推定量
2. k-近傍回帰法
3. B-スプライン

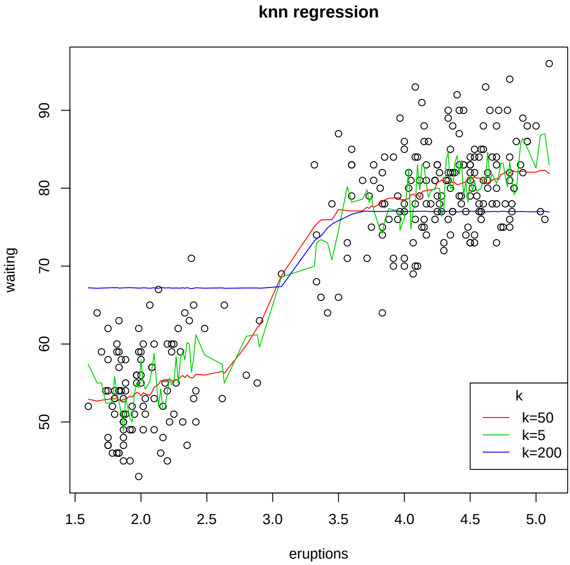
k-近傍回帰法

- 点 x の k-近傍点 $(X_{(1)}, Y_{(1)}, \dots, X_{(k)}, Y_{(k)})$ を取ってくる.
- 次の式で $f(x)$ を推定:

$$\hat{f}(x) = \frac{1}{k} \sum_{j=1}^{k} Y_{(j)}.$$

導出は k-近傍判別と同様.

k-近傍回帰法の実験


knn regression
waiting
eruptions
90
80
70
60
50
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
k
k=50
k=5
k=200

R のコード

```
library(FNN)
knn.reg.res <- knn.reg(train, test = NULL, y, k = 3,
           use.all = FALSE, algorithm=c("VR", "brute",
           "kd_tree", "cover_tree"))
```

- train: 訓練データの x を格納した行列かデータフレーム
- test: テストデータの x
- y: 訓練データの y
- k: 最近傍点の数
- algorithm: 最近傍点を求めるアルゴリズムの指定
- use.all: VR の時のみに有効．タイを全て用いて k-近傍法．指定しなければタイの中からランダムに選択して k-近傍を構成．

knn.reg.res には k,n,pred,residuals などが格納されている．pred に予測値が格納されている．

構成

1 Nadaraya-Watson 推定量

2 k-近傍回帰法

3 B-スプライン

スプライン回帰

ノンパラメトリック回帰の基本的構造:

$$f(x) = \sum_{j=1}^{q} \alpha_j B_j(x).$$

$B_j(x)$ はある非線形な基底関数．ここでは，「**B-スプライン基底**」を考える．

B-スプライン基底

B-スプライン基底関数は局所的な多項式関数である．
例えば，３次 B-スプラインの場合，B_k は次のような局所３次多項式である:

$$B_k(x) = \begin{cases} a_k + b_k x + C_k x^2 + d_k x^3 & (t_k \leq x \leq t_{k+4}), \\ 0 & (\text{otherwise}). \end{cases}$$

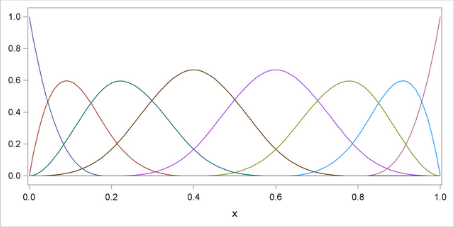
ここで，t_k は節点 (節点) と言う．

- 節点 (節点): $t_1 \leq \cdots \leq t_{q+1}$

３次スプラインの場合，t_k の候補としてサンプル点 X_i を用いて次のようにしたりする:

$$t_1 \leq t_2 \leq t_3 \leq t_4 = X_1 < \cdots < X_n = t_{n+3} \leq t_{n+4} \leq t_{n+5} \leq t_{n+6}.$$

($(X_1, \ldots, X_n)$ の外側に 6 個余分に節点を適当に設定)



B-スプライン基底の具体的な形

係数 a_k, b_k, c_k, d_k の決め方は本講義の範疇を超えるが，j 次 B-スプライン基底は

$$B_k^{(1)}(x) = \begin{cases} 1 & (t_k \leq x \leq t_{k+1}) \\ 0 & (\text{otherwise}) \end{cases},$$

$$B_k^{(j)}(x) = \frac{x - t_k}{t_{k+j} - t_k} B_k^{(j-1)}(x) + \frac{t_k - x}{t_{k+j+1} - t_{k+1}} B_{k+1}^{(j-1)}(x),$$

なる漸化式で決まる．$j = 3$ の時に 3 次 B-スプライン基底を得る．

要は非線形な関数を局所的に 3 次多項式で近似しましょうということ．

3 次 B-スプラインを 3 次-スプライン，キュービック-スプラインと呼んだりする．

回帰係数の決定と平滑化

以後，3 次スプライン基底考え，各データ点 X_i は節点になっているものとする．
節点が X_i であるような 3 次スプライン基底を $B_i(x)$ $(i = 1, \ldots, n)$ と書き直し，
$B_{n+1}(x) = x$, $B_{n+2}(x) = 1$ とする．
この合計 $n + 2$ 個の基底を用い，次のような関数を構成する:

$$f(x;\alpha) = \sum_{i=1}^{n} \alpha_i B_i(x) + \alpha_{n+1} x + \alpha_{n+2}.$$

$\alpha \in \mathbb{R}^{n+2}$ をデータから推定したい．3 次スプラインでは次のようにして推定する．

$$\min_{\alpha \in \mathbb{R}^{n+2}} \sum_{i=1}^{n} (Y_i - f(X_i;\alpha))^2 + \underbrace{\lambda \int_{X_1}^{X_n} \left(\frac{\mathrm{d}^2 f(x;\alpha)}{\mathrm{d}x^2} \right)^2 \mathrm{d}x}_{\text{正則化項}} .$$

※ 正則化項を加えることで，推定した関数が滑らかになるように調整 →平滑化．
これがなければデータに**過適合**してしまう．
※ $\lambda > 0$ は適切に選ぶ必要がある (クロスバリデーション)．
※ リッジ回帰と対応．

二次関数への展開

二乗損失は次のように展開される:

$$\sum_{i=1}^{n}(Y_i - f(X_i;\alpha))^2 = \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{n+2}\alpha_j B_j(X_i))^2$$

$$= \sum_{j=1}^{n+2}\sum_{j'=1}^{n+2}\alpha_j\alpha_{j'}\sum_{i=1}^{n}(B_j(X_i)B_{j'}(X_i)) - 2\sum_{j=1}^{n+2}\alpha_j\sum_{i=1}^{n}Y_iB_j(X_i) + Y^\top Y$$

$$=: \alpha^\top \mathbf{B}^\top \mathbf{B}\alpha - 2Y^\top \mathbf{B}\alpha + Y^\top Y.$$

正則化項は次のように展開される:

$$\int_{X_1}^{X_n}\left(\frac{\mathrm{d}^2 f(X_i;\alpha)}{\mathrm{d}x^2}\right)^2 \mathrm{d}x = \int_{X_1}^{X_n}\left(\sum_{j=1}^{n+2}\alpha_j\frac{\mathrm{d}^2 B_j(x)}{\mathrm{d}x^2}\right)^2 \mathrm{d}x$$

$$= \sum_{j=1}^{n+2}\sum_{j'=1}^{n+2}\alpha_j\alpha_{j'}\int_{X_1}^{X_n}\frac{\mathrm{d}^2 B_j(x)}{\mathrm{d}x^2}\frac{\mathrm{d}^2 B_{j'}(x)}{\mathrm{d}x^2}\mathrm{d}x$$

$$=: \sum_{j=1}^{n+2}\sum_{j'=1}^{n+2}\alpha_j\alpha_{j'}G_{j,j'} = \alpha^\top G\alpha.$$

二次式の最小化

$$\mathbf{B}_{i,j} = B_j(X_i),\ \ G_{j,j'} = \int_{X_1}^{X_n} \frac{\mathrm{d}^2 B_j(x)}{\mathrm{d}x^2} \frac{\mathrm{d}^2 B_{j'}(x)}{\mathrm{d}x^2} \mathrm{d}x,$$

として，

$$\begin{aligned} & \min_{\alpha \in \mathbb{R}^{n+2}} \sum_{i=1}^{n} (Y_i - f(X_i; \alpha))^2 + \lambda \int_{X_1}^{X_n} \left(\frac{\mathrm{d}^2 f(x; \alpha)}{\mathrm{d}x^2} \right)^2 \mathrm{d}x \\ \Leftrightarrow \quad & \min_{\alpha \in \mathbb{R}^{n+2}} \alpha^\top \mathbf{B}^\top \mathbf{B} \alpha - 2Y^\top \mathbf{B} \alpha + \lambda \alpha^\top G \alpha \\ \Rightarrow \quad & \hat{\alpha} = (\mathbf{B}^\top \mathbf{B} + \lambda G)^{-1} \mathbf{B}^\top Y, \end{aligned}$$

で回帰係数が求まる．

平滑化パラメータの決定: CV, GCV

さて，λ はどう選ばばよいか？$\rightarrow$　例のごとくクロスバリデーション (CV).

$$\mathrm{CV} = \frac{\sum_{i=1}^{n}(Y_i - \hat{f}_{(-i)}(X_i))^2}{n}$$

$\hat{f}_{(-i)}(X_i)$ は (X_i, Y_i) を抜いて推定した 3 次スプライン関数.

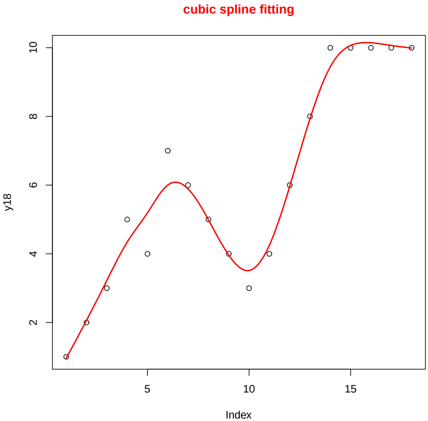
$$\mathrm{GCV} = \frac{\sum_{i=1}^{n}(Y_i - \hat{f}(X_i))^2}{n(1 - \mathrm{Tr}[A(\lambda)]/n)^2},$$

ただし，$A(\lambda) := \mathbf{B}(\mathbf{B}^\top\mathbf{B} + \lambda G)^{-1}\mathbf{B}^\top$. GCV は CV の計算を楽にするために提案されたが，統計的に良い性質があることが知られている.
R (smooth.spline) のデフォルトは GCV.

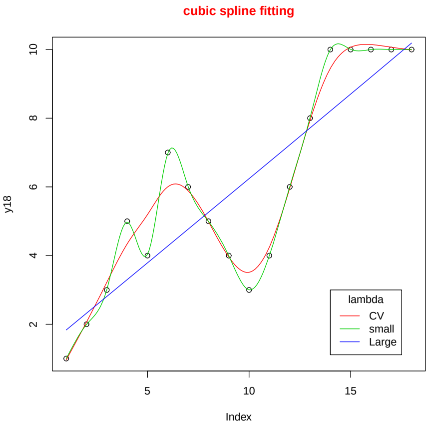
スプラインを実行

```
y18 <- c(1:3, 5, 4, 7:3, 2*(2:5), rep(10, 4))
artdata <- data.frame(x=1:18,y=y18)
s01  <- smooth.spline(artdata)
```

で自動的に λ も GCV で選択.



正則化パラメータの影響



正則化パラメータが小さいと過適合，大きすぎると線形回帰になる．
ちょうど良いパラメータは GCV で選ぶ．

```
s02  <- smooth.spline(artdata, spar = 0.02)
s03  <- smooth.spline(artdata, spar = 1)
```

スプラインを実行: gam

同様のことが mgcv パッケージに入っている gam という関数でも可能.

```
gam.s01 <- gam(y~s(x),data=artdata)
```

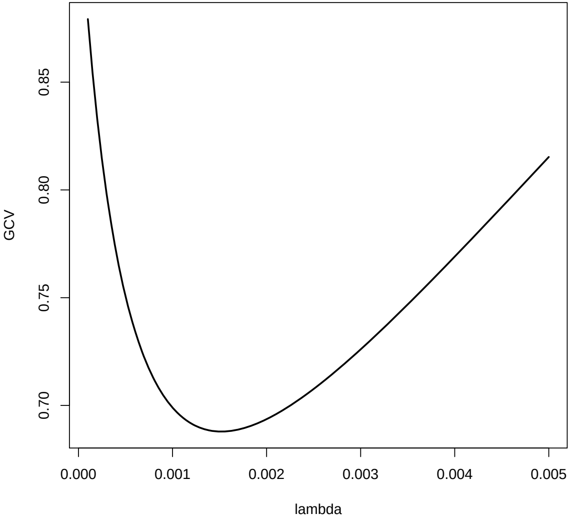
s(x) と書くことでスプラインを使うことを指定．細かい次数や節点の設定も可能.
これで勝手に GCV で正則化パラメータを選んでフィッティングしてくれる.

gam の GCV 値

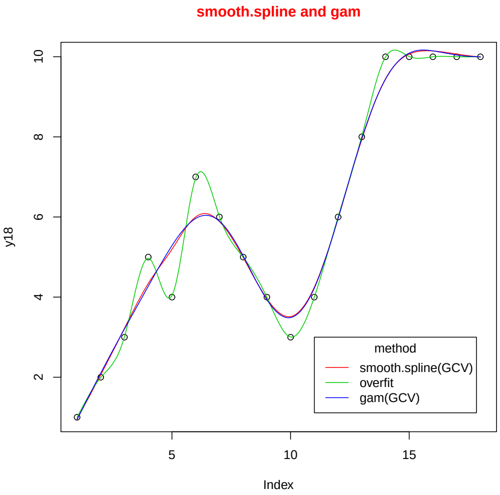
gam を使って，GCV 値を計算してみる．

```
sp<-seq(from=0.0001,to=0.005,length=100)
GCV_art<-numeric()
for(i in 1:length(sp)){
    g.m<-gam(y~s(x),sp=sp[i],data=artdata)
    GCV_art[i]<-g.m$gcv.ubre
}
plot(sp,GCV_art,type="l",lwd=2,xlab="lambda",ylab="GCV")
```

gam の GCV 値



smooth.spline と gam の比較



ほぼ同じ結果

まとめ

ノンパラメトリックな回帰手法を紹介した．

1. カーネル回帰，Nadaraya-Watson 推定量
2. k-近傍法
3. スプライン回帰

バンド幅，k, 正則化定数を CV などを用いてうまく調整する必要があった．

講義情報ページ

http://www.is.titech.ac.jp/~s-taiji/lecture/2015/dataanalysis/dataanalysis.html