统计学习第三章

贾金柱

OCT 24, 2017

讲课老师

- 主讲人： 贾金柱
  - Email: jzjia@math.pku.edu.cn
- 助教：肖一君
  - Email: xiaoyijun1994@126.com

第三章 Kernal Methods

- 课程目标
  - 基底展开
  - 分段多项式和样条
  - 非线性回归和非线性 Logistic Regression
  - 正则化方法
  - R 做数据分析

简介

- 统计决策告诉我们，在L2 Loss 下，预测Y 最好的函数是 $E(Y|X)$
- 线性模型，假设 $E(Y|X) = X\beta$.
- 许多实际问题，可能并不是线性的

简介

- 如何用非线性的函数来表示 $E(Y|X)$?
- 可以考虑加入原始数据的非线性变化
- 定义 $h_m(X) : \mathbb{R}^p \to \mathbb{R}$, 它表示将一个$p$ 维的观测向量转化成一个新的 featur/predictor.
- 这样，我们重新定义

$$E(Y|X) = \sum_{m=1}^{M} h_m(X)\beta_m.$$

- $h_m(X), m = 1, 2, \ldots, M$ 称为 basis.

简介

函数 $f(X) = \sum_{m=1}^{M} h_m(X)\beta_m$ 称为线性基底展开（linear basis expansion）。

这样做的好处是：一旦 $h_m(X)$ 这个基底函数确定下来，把它们当成新的预测变量，又回到了线性回归。

简介

常见的basis 有

- $h_m(X) = X_m, m = 1, 2, \ldots, p$, linear model
- $h_m(X) = X_j^2$ or $h_m(X) = X_j X_k$ quadratic model
- polynomial model
- $h_m(X) = \log(X_j), \sqrt{X_j}, \ldots$
- $h_m(X) = I(L_m \leq X_m \leq U_m)$, piece-wise constant model

分段多项式（Piecewise Polynomials） 和样条（Splines）

我们首先考虑一维情形的非线性回归

最简单的分段多项式，是0阶多项式（常数）。它的基底可以表示为

$$h_1(X) = I(X < \xi_1), h_2(X) = I(\xi_1 \le X < \xi_2), h_3(X) = I(\xi_2 \le X).$$

对于模型 $Y = f(X) + \epsilon = \sum_{m=1}^{3} \beta_m h_m(X)$ 来讲，如果参数 β_m 的最小二乘估计，就是第m 段 Y 的平均值。

分段多项式（Piecewise Polynomials） 和样条（Splines）

再考虑分段1阶多项式（线性函数）

有两种定义方法：连续函数和不连续函数

不连续函数： basis

$$h_1(X) = I(X < \xi_1), h_2(X) = I(\xi_1 \le X < \xi_2), h_3(X) = I(\xi_2 \le X),$$

以及

$$h_4(X) = XI(X < \xi_1), h_5(X) = XI(\xi_1 \le X < \xi_2), h_6(X) = XI(\xi_2 \le X)$$

需要估计6个参数

分段线性函数

连续情形，需要有两个约束：在两个间断点处，函数值一样。

这样就只要估计4个参数。

basis 可以用这四个

$$h_1(X) = 1, h_2(X) = X, h_3(X) = (X - \xi_1)_+, h_4(X) = (X - \xi_2)_+$$

分段多项式

仍以3个节点的分段多项式为例，这次我们考虑3次多项式。

- 没有任何约束 （$4 \times 3 = 12$ 个参数）
- 连续 （$4 \times 3 - 2 = 10$ 个参数）
- 一节连续 ($4 \times 3 - 2 - 2 = 8$ 个参数)
- 二节连续 ($4 \times 3 - 2 - 2 - 2 = 6$ 个参数) [cubic spline]

$$h_1(X) = 1, h_2(X) = X, h_3(X) = X^2$$

$$h_4(X) = X^3, h_5(X) = (X - \xi_1)_+^3, h_6(X) = (X - \xi_2)_+^3$$

分段多项式

一般地，具有K个节点的M阶样条具有连续的$M-2$节导数连续，对应的basis是

$$h_j(X) = X^{j-1}, j = 1, 2, \ldots, M$$

$$h_{M+\ell}(X) = (X - \xi_\ell)_+^{M-1}, \ell = 1, 2, \ldots, K.$$

常见的 $K = 1, 2, 4$ 即：分段常数、线性、三次样条。

- R code: bs(x, degree=1, knots = c(0.2, 0.4, 0.6)) 给出 $N \times 4$ 矩阵，对应分段线性函数的基底。

分段多项式

- 拟合分段三次函数：

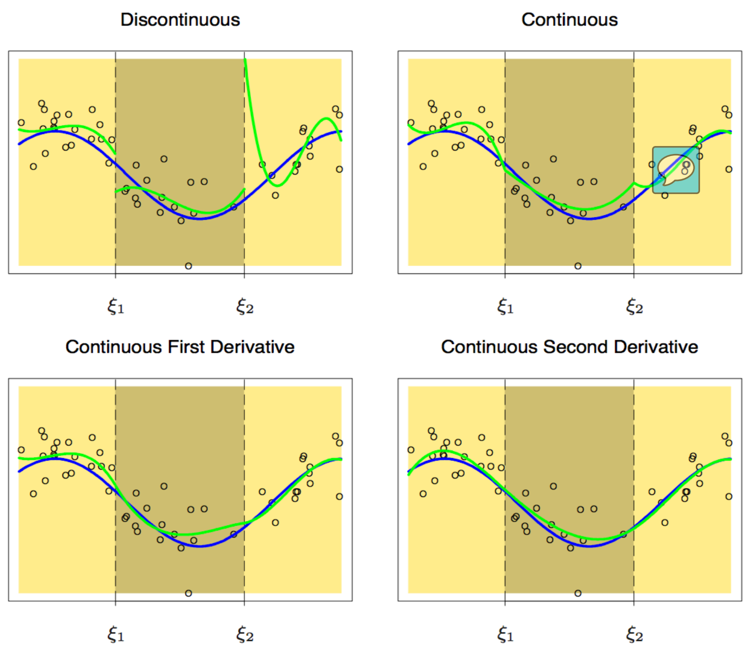


图 1: 拟合分段三次函数

Natural Cubic Splines

- spline 在 boundary 处通常有很大的估计方差：

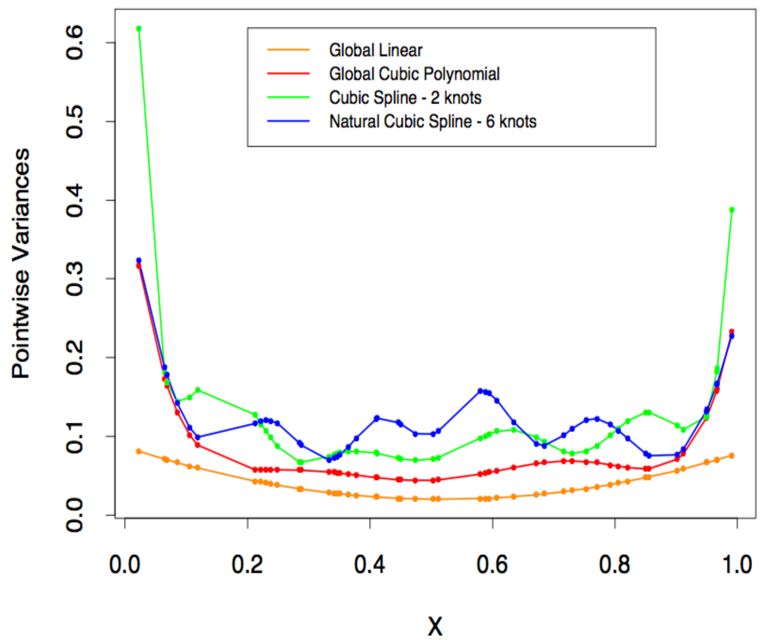


图 2: 各种模型下，估计的误差比较

Natural Cubic Splines

- 为解决在 boundary 处有很大的估计方差这个问题，Natural Cubic Splines，对cubic spline加一个限制：
  - 在boundary nodes 以外的部分，使用线性函数
- 有K个节点的natural cubic spline 可以由 K 个 基底决定。
  - $(K+4)-4$

Basis for Natural Cubic Splines

从 Cubic Spline 的Basis 出发，可以得到，Natrual Cubic Splines 的基底。

注意到，任意的Cubic Spline 可以写成

$$f(x) = \sum_{j=0}^{3} \beta_j x^j + \sum_{k=1}^{K} \theta_k (x - \xi_k)_+^3.$$

为得到Natural Cubic spline, 我们需要加上两个约束：

$$\begin{aligned} \forall x < \xi_1, f'(x) = const \\ \forall x > \xi_k, f'(x) = const \end{aligned}$$

Basis for Natural Cubic Splines

$$f'(x) = \beta_1 + 2\beta_2 x + 3\beta_3 x^2 + \sum_{k=1}^{K} 3\theta_k (x - \xi_k)_+^2$$

由$\forall x < \xi_1, f'(x) = const$，知$\beta_2 = 0, \beta_3 = 0$,
由$\forall x > \xi_K, f'(x) = const$, 知 $\sum_{k=1}^{K} 6\theta_k (x - \xi_k)_+ = 0, \forall x > \xi_K$,

即$\sum_{k=1}^{K} \theta_k = 0, \sum_{k=1}^{K} \theta_k \xi_k = 0.$

-Basis

$$N_1(x) = 1, N_2(x) = x, N_{k+2}(x) = d_k(x) - d_{K-1}(x),$$

$$d_k(x) = \frac{(x - \xi_k)_+^3 - (x - \xi_K)_+^3}{\xi_K - \xi_k}.$$

Example

$$logit(P(Y=1|X)) = \theta_0 + h_1(X_1)^T\theta_1 + \cdots + h_p(X_p)^T\theta_p.$$

- $h_j(X_j) \in \mathbb{R}^{K-1}$ with K knots.
- $\theta_j \in \mathbb{R}^{K-1}$
- We could rewrite the whole model as

$$logit(P(Y|X)) = H\theta.$$

这就是传统的Logistic Regression， 它的预测变量矩阵是H.

- 原问题的变量选择，对应于新的问题的group selection
- 可以使用 group Lasso
- 也可以使用AIC / BIC

Example （发音识别）

- 两组不同的发音数据（频谱分析）：

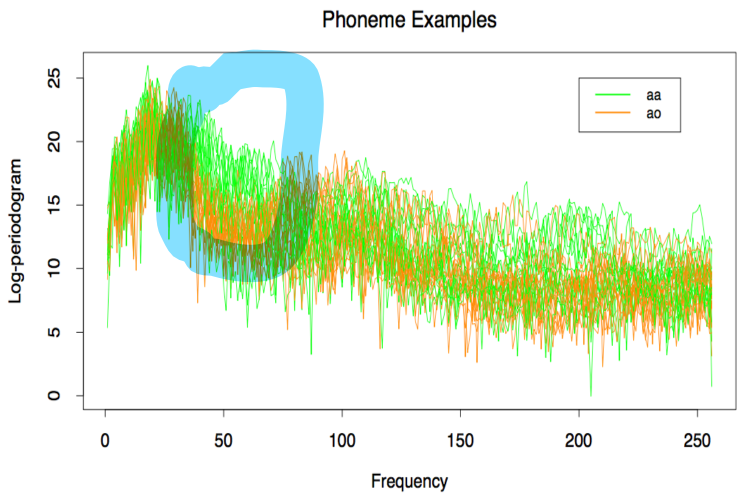


图 3: 两组不同的发音数据

$$logit(Y = 1|X) = X(f)\beta(f) = XH\theta = X^{*}\theta.$$

Example (发音识别)

$$logit(Y = 1|X) = X(f)\beta(f) = XH\theta(f) = X^*\theta.$$

- 两组不同的发音数据（频谱分析）：

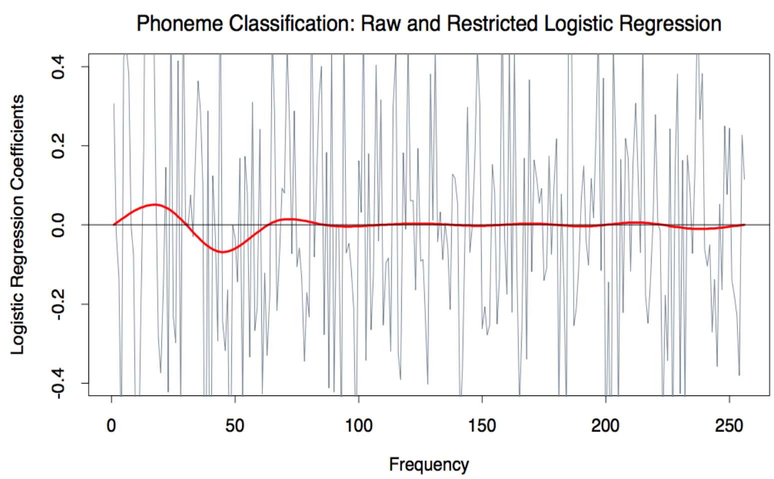


图 4: Logistic 回归系数

Example (发音识别)

	Raw	*spline*
Training error	0.080	0.185
Test erroe	0.255	0.158

Nonparametric Regression and Modelling

- Nonparametric Regression

$$Y = f(X) + \epsilon$$

- Nonparametric Logistic Regression

$$logit(P(Y = 1|X)) = f(X)$$

Generalized Linear Models

- GLM 用来将很多模型统一起来
- 包括： 线性模型、logist 回归、Poisson 回归等
- GLM 三要素：
  - $Y|X$ 的分布族
  - 线性预测 $\eta = X\beta$
  - 连接函数 $g(E(Y)) = \eta = X\beta$

Generalized Linear Models

线性模型的三要素:

- $Y|X = x_i$ 服从均值为μ_i, 方差为σ^2的正态分布
- 线性预测 $\eta_i = x_i^T \beta$
- 连接函数 $g(x) = x$, $E(Y_i) = \eta_i = x_i^T \beta$

Logistic 回归的三要素

- $Y|X = x_i$ 服从均值为 μ_i 的二项分布
- 线性预测 $\eta_i = x_i^T \beta$
- 连接函数 $g(\mu) = \log(\frac{\mu}{1-\mu})$,

$$g(E(Y_i)) = g(\mu_i) = \eta_i = x_i^T \beta$$

$$\mu_i = P(Y_i = 1|X = x_i) = \frac{e^{x_i^T \beta}}{1 + e^{x_i^T \beta}}$$

Poisson 回归的三要素

- $Y|X = x_i$ 服从均值为 μ_i 的Poisson 分布
- 线性预测 $\eta_i = x_i^T\beta$
- 连接函数 $g(\mu) = \log(x)$,

$$g(E(Y_i)) = \log(\mu_i) = \eta_i = x_i^T\beta$$

Generalized Linear Models

- $Y|X = x_i$ 服从扩展的指数分布族

$$f(y;\theta,\psi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\psi)} + c(y;\psi)\right\},$$

其中，θ 和 ψ 跟 X 有关， $a(\cdot), b(\cdot), c(\cdot)$是三个函数

- 线性预测 $\eta_i = x_i^T\beta$
- 连接函数

$$g(E(Y|X = x_i)) = \eta_i = x_i^T\beta$$

GLM 连接函数的选取

定义. 考虑扩展的指数分布族

$$Y \sim f(y;\theta,\psi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\psi)} + c(y;\psi)\right\},$$

如果连接函数$g(\cdot)$, 满足 $g(E(Y)) = \theta$, 则称之为典型连接函数。

一些典型连接函数

- 典型链接函数：

	Normal	Poisson
Notation	$N(\mu, \sigma^2)$	$Poisson(\mu)$
log-density	$\frac{1}{\sigma^2}(y\mu - \frac{1}{2}\mu^2 - \frac{1}{2}y^2) - \frac{1}{2}\log(2\pi\sigma^2)$	$y\log(\mu) - \mu - \log(y!)$
Range of y	$(-\infty, \infty)$	$0, 1, 2, \ldots, \infty$
Dispersion parameter, ψ	σ^2	1
$a(\psi)$	ψ	ψ
$b(\theta)$	$\theta^2/2$	e^θ
$c(y;\psi)$	$-\frac{1}{2}(\frac{y^2}{\psi} + \log(2\pi\psi))$	$-\log(y)$
$\mu = E(Y)$	θ	e^θ
Canonical link function (CLF) $(\theta = g(\mu))$	identity $(\theta = \mu)$	$\log(\mu)$

	Binomial	Gamma
Notation	$Bino(n, \mu)/n$	$G(\mu, \nu)$
log-density	$n[y\log(\frac{\mu}{1-\mu}) + \log(1-\mu)] + \log\begin{pmatrix} n \\ ny \end{pmatrix}$	$v(-\frac{y}{\mu} - \log(\mu)) + v\log(y) + v\log(v) - \log(\Gamma(v))$
Range of y	$z/n, z \in \{0, 1, 2, \ldots\}$	$(0, \infty)$
ψ	n^{-1}	v^{-1}
$a(\psi)$	ψ	ψ
$b(\theta)$	$\log(1 + e^\theta)$	$-\log(-\theta)$
$c(y;\psi)$	$\log\begin{pmatrix} n \\ ny \end{pmatrix}$	$v\log(vy) - \log(y) - \log(\Gamma(v))$
$\mu = E(Y)$	$\frac{e^\theta}{1+e^\theta}$	$-1/\theta$
CLF	$\log(\frac{\mu}{1-\mu})$	$-\mu^{-1}$

TABLE 1

Common distributions in the extended exponential family and their canonical link functions.

图 5: 典型连接函数

一个性质:

如果

$$Y \sim f(y;\theta,\psi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\psi)} + c(y;\psi)\right\},$$

则

$$E(Y) = \dot{b}(\theta) = \frac{d}{d\theta} b(\theta)$$

$$var(Y) = \ddot{b}(\theta)$$

MLE of GLM

- GLM 的似然函数

$$\prod_{i=1}^{n} \exp\Big\{ \frac{y_i x_i^T \beta - b(x_i^T \beta)}{a(\psi)} + c(y_i; \psi) \Big\}$$

- 对数似然函数

$$\sum_{i=1}^{n} \Big[\frac{y_i x_i^T \beta - b(x_i^T \beta)}{a(\psi)} + c(y_i; \psi) \Big]$$

- MLE

$$\hat{\beta} = \arg\max_{\beta} \sum_{i=1}^{n} \Big[y_i x_i^T \beta - b(x_i^T \beta) \Big]$$

MLE of GLM

$$\hat{\beta} = \arg\max_{\beta} \sum_{i=1}^{n} \left[y_i x_i^T \beta - b(x_i^T \beta) \right]$$

- $b(\theta) = \frac{\theta^2}{2}$ – 线性回归（least squares）
- $b(\theta) = e^{\theta}$ – Poisson 回归
- $b(\theta) = \log(1 + e^{\theta})$ – logistic 回归

MLE of GLM [Iteratively weighted least squares]

$$\hat{\beta} = \arg\max_{\beta} f(\beta) := \arg\max_{\beta} \sum_{i=1}^{n} \left[y_i x_i^T \beta - b(x_i^T \beta) \right]$$

- 一阶导数

$$\dot{f}(\beta) = \sum_{i=1}^{n} \left[y_i x_i - x_i \dot{b}(x_i^T \beta) \right] = X^T (Y - \hat{Y}(\beta))$$

- 二阶导数

$$\ddot{f}(\beta) = -\sum_{i=1}^{n} x_i x_i^T \ddot{b}(x_i^T \beta) := -X^T B X$$

IWLS

$$
\begin{aligned}
\beta^{(t+1)} &= \beta^{(t)} - [\ddot{g}(\beta^{(t)})]^{-1}\dot{g}(\beta^{(t)}) \\
&= \beta^{(t)} + (X^T BX)^{-1} X^T (Y - \hat{Y}^{(t)}) \\
&= (X^T BX)^{-1}[(X^T BX)\beta^{(t)} + X^T (Y - \hat{Y}^{(t)})] \\
&= (X^T BX)^{-1} X^T B[X\beta^{(t)} + B^{-1}(Y - \hat{Y}^{(t)})] \\
&= \arg\min_{\beta} \sum_{i=1}^{n} B_{ii}(\tilde{y}_i - x_i^T \beta)^2.
\end{aligned}
$$

其中，

$$
\tilde{Y} = X\beta^{(t)} + B^{-1}(Y - \hat{Y}^{(t)}).
$$

Smoothing splines

前面讲的spline regression，有两个重要问题：选取几个knots, knots 该如何选取?

Smoothing splines 将不指定哪些点做knots，它使用正则项（惩罚项）来控制拟合曲线的复杂度。

具体地，考虑所有的拥有二阶连续导数的函数 $f(x)$, 我们从中选取最小化如下的带惩罚的残差平方和“

$$RSS(f, \lambda) = \sum_{i=1}^{N} \{y_i - f(x_i)\}^2 + \lambda \int \{f''(t)\}^2 dt.$$

Smoothing splines (续)

两个特例：

- $\lambda = 0$: RSS 可以取到0，此时曲线 $f(x)$ overfits the data
- $\lambda = \infty$: 此时 $f''(x) = 0$, 最好的拟合曲线是直线–一次函数

这两个特例，从非常 rough 的曲线到非常smooth 的曲线，smooth spline 的目标是选取合适的λ, 使得曲线既可以很好地拟合data，同时又比较smooth。

Smoothing splines （续）

- smooth spline 是定义在一个无穷维的泛函空间的最优化
- 这个泛函空间叫做 Sobolev space
- 一个重要性质： for $\lambda \in (0, \infty)$, smooth spline 的解是一个natural cubic spline with knots at the unique values of $x_i, i = 1, 2, \ldots, N$.

Smoothing splines (续)

- 我们通过证明以下几个事实，来证明 smooth spline 的解是一个 natural cubic spline。

（1）. 对于任意的 n 个knots, $a < x_1 < x_2 <\ldots < x_n < b$ 以及任意指定的N个实数$z_1, z_2, \ldots, z_N$. 总存在唯一的 NCS（natural cubic spline） $g(x)$, 使得

$$g(x_i) = z_i.$$

（2）. 设 $g(x)$ 是一个 NCS，对于任意的拥有二阶连续可微的函数 $\tilde{g}(x)$, 且 $\tilde{g}(x_i) = g(x_i)$, 有

$$\int_a^b \{g''(t)\}^2 dt \leq \int_a^b \{\tilde{g}''(t)\}^2 dt$$

Smoothing splines (续)

（3）. 由 （1）和 （2） 知，如果 $\tilde{g}(x)$ 是 smooth spline 的一个解，则通过构造一个CNS $g(x)$, 使得 $g(x_i) = \tilde{g}(x_i)$ 得到一个CNS解。

（4）. smooth spline 的解是唯一的。

Smoothing splines (续)

（1） 的证明：

我们知道 n 个 knots 的 natural cubic spline 可以写成 n 个基底的线性组合： $g(x) = \sum_{i=1}^{n} N_i(x)\beta_i$. 由

$$z_k = g(x_k) = \sum_{i=1}^{n} N_i(x_k)\beta_i,$$

可以得到一个线性方程组

$$z = N_{n\times n} \times \beta,$$

其中 $\beta \in \mathbb{R}^{n\times 1}$, $N_{n\times n} \in \mathbb{R}^{n\times n}$ 是一个$n \times n$的满秩矩阵。 于是系数 β 可以唯一确定。

Smoothing splines (续)

（2） 的证明：

令 $h(x) = \tilde{g}(x) - g(x)$.

$$
\begin{aligned}
\int_a^b g''(x)h''(x)dx &= \int_a^b g''(x)dh'(x) \\
&= g''(x)h'(x)|_a^b - \int_a^b h'(x)g'''(x)dx \\
&= 0 - \int_a^{x_1} h'(x)g'''(x)dx \\
&- \sum_{k=1}^{n-1} \int_{x_k}^{x_{k+1}} h'(x)g'''(x)dx - \int_{x_n}^b h'(x)g'''(x)dx \\
&= 0
\end{aligned}
$$

Smoothing splines (续)

（2） 的证明：

由上页证明，知

$$\int_a^b g''(x)\tilde{g}''(x)dx = \int_a^b \{g''(x)\}^2 dx.$$

注意到

$$\int_a^b g''(x)\tilde{g}''(x)dx \le \int_a^b g''(x)^2 dx \int_a^b \{\tilde{g}''(x)\}^2 dx$$

所以

$$\int_a^b \{g''(t)\}^2 dt \le \int_a^b \{\tilde{g}''(t)\}^2 dt$$

Smoothing splines (续)

（3）的证明很显然。

（4）的证明：

如果 $\tilde{g}(x)$ 是 smooth spline 的一个解，则通过构造一个CNS $g(x)$, 使得 $g(x_i) = \tilde{g}(x_i)$, 可以 得到一个CNS解。由（2） 知

$$\int_a^b \{g''(t)\}^2 dt = \int_a^b \{\tilde{g}''(t)\}^2 dt,$$

此时 $g''(x) = \lambda \tilde{g}''(x), \lambda \geq 0$. 注意 $\int_a^b \{g''(t)\}^2 dt = \int_a^b \{\tilde{g}''(t)\}^2 dt$, 从而 $\lambda = 1$. 于是 $g'(x) - \tilde{g}'(x) = const, \forall x$, 于是存在常数$a, b$, 使得 $g(x) - \tilde{g}(x) = ax + b, \forall x$, a 和 b 必然是0. 这说明，smooth spline 的解必然是CNS。

Smoothing splines (续)

（4）的证明（续）： 前面已证 smooth spline 的解必然是 CNS。 因此，它的解具有如下形式

$$f(x) = \sum_{i=1}^{n} N_i(x)\theta_i.$$

把它带入到目标函数，

$$RSS(f,\lambda) = \sum_{i=1}^{N} \{y_i - f(x_i)\}^2 + \lambda \int \{f''(t)\}^2 dt.$$

$$RSS(f,\lambda) = \sum_{k=1}^{N} \{y_k - \sum_{k=1}^{n} N_i(x_k)\theta_i\}^2 + \lambda \int \{\sum_{i=1}^{n} N_i''(x)\theta_i\}^2 dt.$$

Smoothing splines (续)

（4）的证明（续）：

将上式写成矩阵形式：

$$RSS(f, \lambda) = \|Y - N\theta\|_2^2 + \lambda\theta^T\Omega\theta,$$

其中 $N \in \mathbb{R}^{n\times n}$ with $N_{ik} = N_i(x_k)$ and $\Omega \in \mathbb{R}^{n\times n}$ with $\Omega_{ik} = \int N_i''(x)N_k''(x)dx$.

易知，θ 的解是

$$\hat{\theta} = (N^T N + \lambda\Omega)^{-1}N^T y.$$

The fitted smooth spline is

$$\hat{f}(x) = \sum_{j=1}^{n} N_j(x)\hat{\theta}_j.$$

tuning parameter selection

- bias and variance trade-off

$$
\begin{aligned}
EPE(\hat{f}_\lambda) &:= E(Y - \hat{f}_\lambda(X))^2 \\
&= E[Y - EY + f(X) - \hat{f}_\lambda(X)]^2 \\
&= E(Y - EY)^2 + E[f(X) - \hat{f}_\lambda(X)]^2 \\
&= E(Y - EY)^2 + E[f(X) - E(\hat{f}_\lambda(X)) + E(\hat{f}_\lambda(X)) - \hat{f}_\lambda(X)]^2 \\
&= var(Y) + Bias^2(f_\lambda(X)) + var(f_\lambda(X)).
\end{aligned}
$$

tuning parameter selection (续)

- CV (Cross Validation)

$$
\begin{aligned}
CV(\hat{f}_\lambda) &:= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}_\lambda^{(-i)}(x_i))^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\Big(\frac{y_i - \hat{f}_\lambda(x_i)}{1 - S_\lambda(i,i)}\Big)^2,
\end{aligned}
$$

其中， S_λ 称为 smoother matrix defined as

$$
S_\lambda := N(N^T N + \lambda\Omega_N)^{-1}N^T
$$

Degree of Freedom (自由度)

- 考虑简单线性回归， $Y = X\beta + \epsilon$, 参数β 的最小二乘估计是 $\hat{\beta} = (X^T X)^{-1} X^T Y$, fitted values are

$$\hat{Y} = X(X^T X)^{-1} X^T Y := HY.$$

$$tr(H) = tr(X(X^T X)^{-1} X^T) = tr((X^T X)^{-1} X^T X) = p.$$

- 对于 smooth spline， fitted values are

$$\hat{Y} = N\hat{\theta} = N(N^T N + \lambda\Omega_N)^{-1} N^T Y := S_\lambda Y$$

$df_\lambda := trace(S_\lambda)$ 称为smooth spline 的有效自由度，简称自由度。

- 当$\lambda = 0$时，$df_\lambda = n$, 所有的knots 都是有效的。

Degree of Freedom (自由度)

注意到

$$
\begin{aligned}
S_\lambda &:= N(N^T N + \lambda\Omega_N)^{-1} N^T \\
&= [N^{-T}(N^T N + \lambda\Omega_N) N^{-1}]^{-1} \\
&= [I + \lambda K]^{-1},
\end{aligned}
$$

where $K = N^{-T}\Omega_N N^{-1}$, 它不依赖于$Y$,仅仅依赖于$x$.

对 K 做特征值 分解，有 $K = UDU^T$, 则

$$
S_\lambda = [UU^T + \lambda UDU^T]^T = U(I + \lambda D)^{-1} U^T.
$$

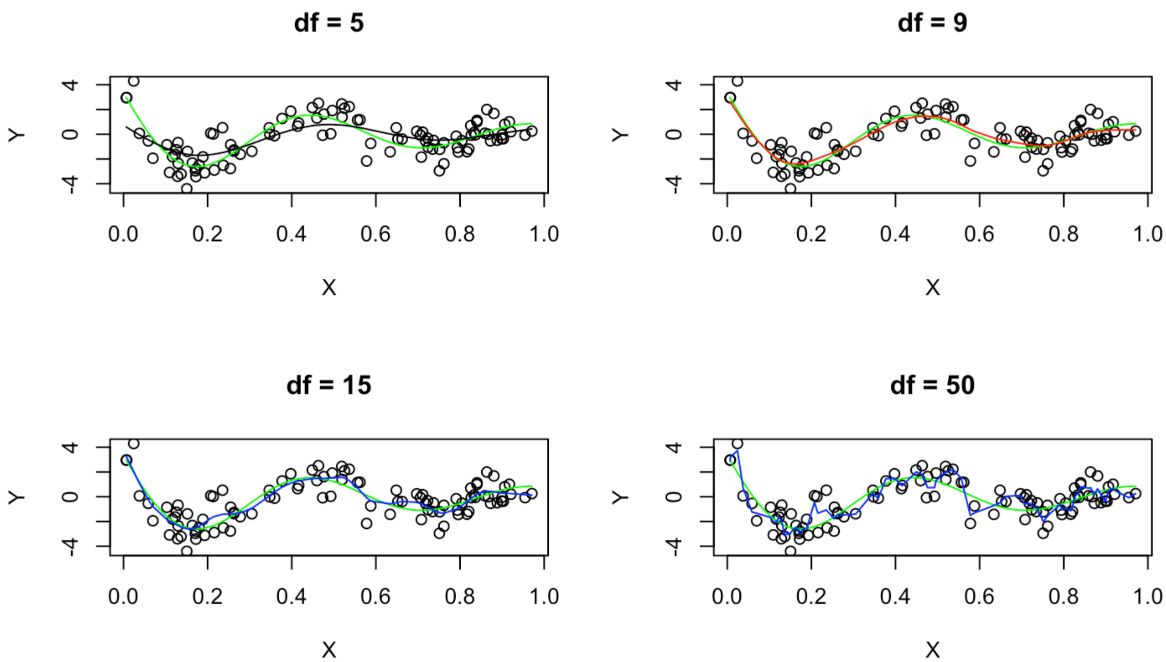
所以,

$$
df_\lambda = \sum_{k=1}^{n} \frac{1}{1 + \lambda d_k}.
$$

Experiments

- Model Settings
  - $X \sim U[0, 1]$
  - $\epsilon \sim N(0, 1)$
  - $Y = f(X) + \epsilon$
  - $f(X) = \frac{sin(12(X+0.2))}{X+0.2}$,
  - training samples are i.i.d from the model with sample size $n = 100$.

Experiments



Experiments

- get the bias and variances

$\hat{f} = S_\lambda y$, so

$$Cov(\hat{f}) = S_\lambda cov(y) S_\lambda^T = S_\lambda S_\lambda^T.$$

$$Bias(\hat{f}) = f - E(\hat{f}) = f - S_\lambda f.$$

Homework

- Due: Nov 8.

1. Reproduce Figure 5.3
2. Reproduce Figure 5.9
3. Ex. 5.5. Write a program to classify the phoneme data using a quadratic dis- criminant analysis (Section 4.3).
4. Ex. 5.13

multi-dimensional splines

- generalize one-dimensional smooth spline to multidimensional case

$$\min_f \sum_{i=1}^{n} \{y_i - f(x_i)\}^2 + \lambda J(f).$$

For two-dimensional problems,

$$J(f) = \iint_{\mathbb{R}^2} \left[\left(\frac{\partial^2 f(x)}{\partial x_1^2}\right)^2 + 2\left(\frac{\partial^2 f(x)}{\partial x_1 \partial x_2}\right)^2 + \left(\frac{\partial^2 f(x)}{\partial x_2^2}\right)^2 \right] dx_1 dx_2$$

multi-dimensional splines

- The above optimization leads to thin-plate spline with the solution having the following form

$$f(x) = \beta_0 + \beta^T x + \sum_{j=1}^{n} \alpha_j h_j(x),$$

where

$$h_j(x) = \|x - x_j\|_2^2 \log(\|x - x_j\|_2).$$

multi-dimensioanl splones

- additive spline models

$f(x) = \alpha + f_1(X_1) + \ldots + f_d(X_d)$ and

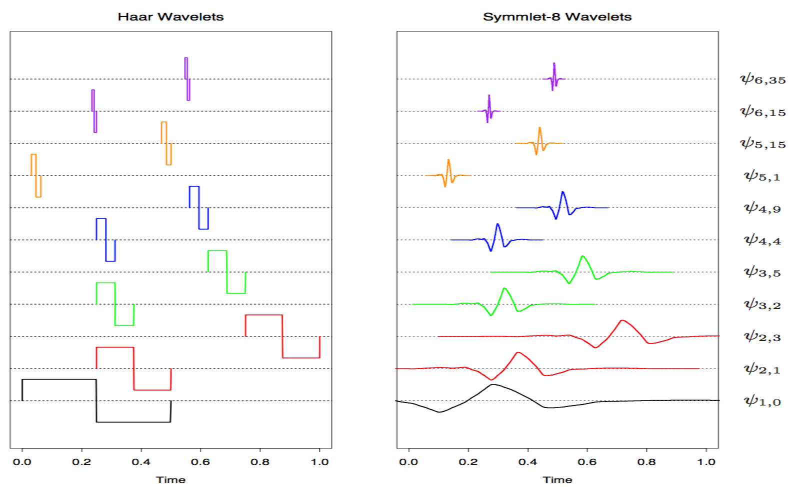
$$J(f) = \sum_{j=1}^{d} \int [f_j''(t)]^2 dt$$

- ANOVA spline decompositions

$$f(X) = \alpha + \sum_j f_j(X_j) + \sum_{j<k} f_{jk}(X_j, X_k) + \cdots$$

wavelet smoothing

- smooth spline tends to fit smooth curves
- for curves with smooth part and bumpy part, wavelet smoothing is more appropriate
- wavelet basis functions：



图：wavelet basis functions

computation for splines

spline basis for piecewise polynomial function could be computed recursively from order 1 to any order.

用 $B_{i,m}$ 来代表 order 为 m 的 polynomial spline 的第 i 个 base，则他们可以通过如下的递推公式快速获得：

$$B_{i,1}(x) = I_{\tau_i \leq x < \tau_{i+1}}$$

$$B_{i,m}(x) = \frac{x - \tau_i}{\tau_{i+m-1} - \tau_i} B_{i,m-1}(x) + \frac{\tau_{i+m} - x}{\tau_{i+m} - \tau_{i+1}} B_{i+1,m-1}(x)$$

Kernel Smoothing method

- 对于给定的任意点 x_0, 我们使用local fit 的方法，得到一个局部估计
- 很多局部估计放在一起，得到一个smooth 的估计函数 $\hat{f}(X)$
- 局部估计，要使用权重，这个权重就是 Kernel
- 这里的Kernel 主要是指用于指定localization 的weight function

一维 Kernel smoothers

- 首先看一下 KNN （最简单的 kernel smoothing method）

$$\hat{f}(x) = Ave(y_i | x_i \in N_k(x)).$$

- KNN 的特点：
  - 是条件期望 $E(Y|X)$ 的很好的估计
  - 简单
  - 但是，它的缺点是，估计的曲线不光滑
- KNN 估计可以重新表述：

$$\hat{f}(x) = \frac{\sum_{i=1}^{N} K(x_0, x_i) y_i}{\sum_{i=1}^{N} K(x_0, x_i)},$$

其中，$K(x_0, x_i) = 1, if\, x_i$ 落入了x_0的小邻域。

一维 Kernel smoothers

- 选取合适的Kernel function， 可以得到更smooth的估计曲线
- Epanechnikov quadratic kernel:

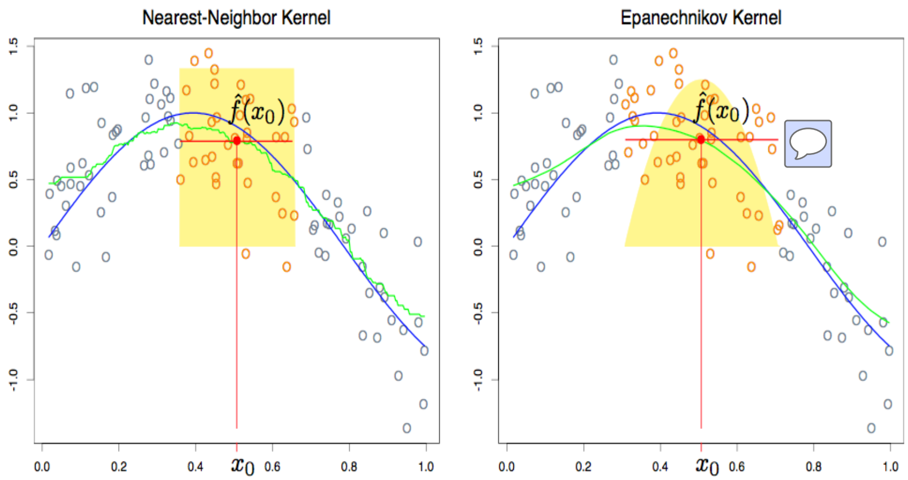
$$K_\lambda(x_0, x) := D(\frac{|x - x_0|}{\lambda}),$$

with

$$D(t) = \begin{cases} \frac{3}{4}(1 - t^2) & if\ |t| \leq 1 \\ 0 & |t| > 1 \end{cases}$$

一维 Kernel smoothers

- KNN 和 Epanechnikov quadratic kernel：



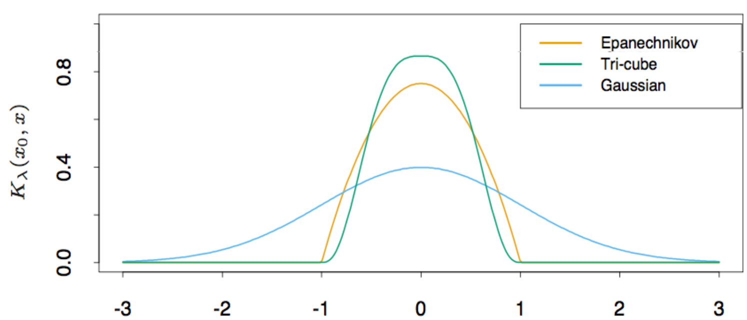
图：KNN 和 Epanechnikov quadratic kernel 的比较

一维 Kernel smoothers

- 常见的Kernel function
  - Epanechnikov quadratic kernel
  - tri-cube

$$D(t) = \begin{cases} (1 - |t|^3)^3 & if\ |t| \leq 1, \\ 0 & if\ |t| > 1 \end{cases}$$

  - Gaussian kernel



图：几种常见的Kernels

Local linear Models

- local constant (or local weighted average) has higher bias at the boundary
- the bias could be reduced by using local linear estimator
- Local linear estimator could be obtained via local weighted least squares

$$\min_{\alpha(x_0),\beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i)[y_i - \alpha(x_0) - \beta(x_0)x_i]^2.$$

Denote $b(x)^T = (1, x)$, $B \in \mathbb{R}^{N\times 2}$ with the ith row $(1, x_i)$, then

$$\hat{f}(x) = b(x_0)^T(B^TW(x_0)B)^{-1}B^TW(x_0)y = L^Ty = \sum_{i=1}^{n} l_i(x_0)y_i$$

Local polynomial models

$$\min_{\alpha(x_0),\beta_j(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i)[y_i - \alpha(x_0) - \sum_{j=1}^{d} \beta_j(x_0) x_i^j]^2.$$

- Local maximum likelihood

$$\max_{\theta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i) \log f(x_i; \theta(x_0))$$

Kernel Density Estimation and Classification

- Kernel Density Estimation (KDE)
  - unsupervised learning
  - could be used for classification

为估计 $f(x)$, 可以考虑在x附近的小邻域 $(x-h, x+h)$,

$$
\begin{aligned}
f(x) * (2h) &\approx \int_{x-h}^{x+h} f(t)dt \\
&= F(x+h) - F(x-h) \\
&\approx F_n(x+h) - F_n(x-h)
\end{aligned}
$$

Kernel Density Estimation

因此，可以使用 $\frac{F_n(x+h)-F_n(x-h)}{2h}$ 来估计 $f(x)$, 即

$$\begin{aligned}\hat{f}_n(x) = & \frac{F_n(x+h)-F_n(x-h)}{2h} \\ = & \frac{\#\{i : x_i \in (x-h, x+h)\}}{2nh} \\ = & \frac{\#\{i : |x-x_i|/h \leq 1\}}{2nh} \\ = & \frac{1}{nh}\sum_{i=1}^{n} K_0(\frac{x-x_i}{h})\end{aligned}$$

这里

$$K_0(\frac{x-x_i}{h}) = \frac{1}{2} I_{|\frac{x-x_i}{h}|\leq 1}$$

Kernel Density Estimation

更一般的KDE 是：

$$\hat{f}_X(x_0) = \frac{1}{N\lambda}\sum_{i=1}^{N} K_\lambda(x_0, x_i) := \frac{1}{N\lambda}\sum_{i=1}^{N} K(\frac{x_0 - x_i}{\lambda}).$$

只要$K_\lambda(x; x_i)$满足如下条件：

$$K(u) \geq 0, \int_{-\infty}^{\infty} K(u)du = 1$$

.

$$\int_{-\infty}^{\infty} uK(u)du = 0, \quad \int_{-\infty}^{\infty} u^2 K(u)du = \sigma_K^2 < \infty$$

.

Kernel Density Estimation

上述估计称为核估计，$K_\lambda(x;x_i)$称为核函数，λ称为核函数的宽度。 常用的核函数有 Gaussian 密度函数等。

性质：假设 $f(x)$ 光滑,且任意的 $x_1 < x_2$, 有 $|f(x_1) - f(x_2)| \le L|x_1 - x_2|$. 假设 $K(u) \le M$. 则存在常数 $c_1 > 0$ 和 $c_2 > 0$, 使得

- $bias(\hat{f}(x)) \le c_1\lambda$
- $var(\hat{f}(x)) \le \frac{c_2}{N\lambda^2}$
- $MSE(\hat{f}(x)) \approx N^{-1/2}$.

Kernel Density Estimation

$$
\begin{aligned}
E(\hat{f}(x)) - f(x) = & \int \frac{1}{\lambda} K(\frac{x-u}{\lambda}) f(u) du - f(x) \\
= & \int K(t) f(x - \lambda t) dt - f(x) \\
= & \int K(t)[f(x - \lambda t) - f(x)] dt
\end{aligned}
$$

所以,

$$
|E(\hat{f}(x)) - f(x)| \leq \lambda \int K(t)|t| dt := c_1 \lambda.
$$

Kernel Density Estimation

$$
\begin{aligned}
var(\hat{f}(x)) \leq & \int \frac{1}{N\lambda^2} K^2(\frac{x-u}{\lambda}) f(u) du \\
\leq & \; c_2 \int \frac{1}{N\lambda^2} f(u) du \\
= & \; \frac{c_2}{N\lambda^2}
\end{aligned}
$$

Fianlly,

$$
MSE(\hat{f}(x)) \leq c_1^2 \lambda^2 + \frac{c_2}{N\lambda^2}
$$

当 $\lambda \propto N^{-1/4}$ 时，$MSE(\hat{f}(x)) \approx N^{-1/2} \to 0$.

Kernel Density Classification

$$\hat{Pr}(G = j|X = x_0) = \frac{\hat{\pi}_j \hat{f}_j(x_0)}{\sum_{k=1}^{J} \hat{\pi}_j \hat{f}_k(x_0)}.$$

- for high-dimensional problems, naive bayes could be used.

$$\hat{f}(X_1, X_2, \ldots, X_p) = \prod_j \hat{f}(X_j).$$