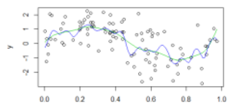
第8章 非参数回归
参考:王星2014 非参数统计chap8



王 星
办公电话:86-10-82500167
电子邮箱:wangxingwisdom@126.com

大 纲

- 核光滑回归
- 局部多项式回归
- 稳健回归
- *K近邻回归
- *正交序列回归
- *B-Spline

Parametric & partial parametric

- Parametric approach: $m(.)$ is known and smooth. It is fully described by a finite set of parameters, to be estimated. Easy interpretation. For example, a linear model:

$$y_i = x_i'\beta + \varepsilon_i, \qquad i = 1, \cdots, N$$

- Nonparametric approach: $m(.)$ is smooth, flexible, but unknown. Let the data determine the shape of $m(.)$. Difficult interpretation.

$$y_i = m(x_i) + \varepsilon_i, \qquad i = 1, \cdots, N$$

- Semi-parametric approach: $m(.)$ have some parameters -to be estimated-, but some parts are determined by the data.

$$y_i = x_i'\beta + m_z(z_i) + \varepsilon_i, \qquad i = 1, \cdots, N$$

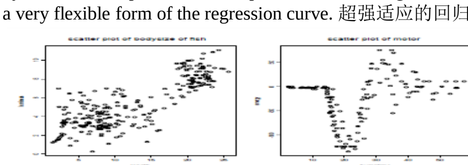
3

1.非参数回归

- The aim of a regression analysis is to produce a reasonable analysis to the unknown response function m, where for n data points (X_i, Y_i), the relationship can be modeled as

$$Y_t = m(X_t) + \sigma(X_t)\varepsilon_t$$
$$Y_i = m(X_i) + \varepsilon_i, \quad i = 1, \Lambda, n \qquad (1)$$

- Unlike parametric approach where the function m is fully described by a finite set of parameters, nonparametric modeling accommodate a very flexible form of the regression curve. 超强适应的回归形式



Motivation

- It provides a versatile method of exploring a general relationship between variables，can be used to test for nonlinearity.
  提供更丰富的用于表达变量关系的视角,表达非线性结构
- It gives predictions of observations yet to be made without reference to a fixed parametric model
  不需要在固定的参数形式下获得预测
- It provides a tool for finding spurious observations by studying the influence of isolated points
  提供了一种发现异常观测并研究它可能影响的方法
- It constitutes a flexible method of substituting for missing values or interpolating between adjacent X-values
  面对数据存在缺失或需要对缺失进行相邻插值时，它的适应性很强

5

光滑回归的基本原理

- A reasonable approximation to the regression curve $m(x)$ will be the mean of response variables near a point x. This ***local averaging procedure*** can be defined as

$$\hat{m}(x) = n^{-1}\sum_{i=1}^{n} W_{ni}(x) Y_i \qquad (2)$$

Every smoothing method to be described is of the form (2).

$$W_{hi}(x) = K_h(x - X_i) / \hat{f}_h(x) \qquad (3)$$

where $\hat{f}_h(x) = n^{-1}\sum_{i=1}^{n} K_h(x - X_i)$, and $K_h(u) = h^{-1}K(u/h)$.

Kernel smoothing describes the shape of the weight function $W_{ni}(x)$ by a density function K with a scale parameter that adjusts the size and the form of the weights near x. The *kernel* K is a continuous, bounded and symmetric real function which integrates to 1。

6

Kernel Smoothing核光滑

- The *Nadaraya-Watson estimator* is defined by

$$\hat{m}_h(x) = \frac{\sum_{i=1}^{n} K_h(x - X_i)Y_i}{\sum_{i=1}^{n} K_h(x - X_i)} \qquad (4)$$

均方误差 $d_M(x,h) = E[\hat{m}_h(x) - m(x)]^2$，当 $n \to \infty,\ h \to 0,\ nh \to \infty$，我们有如下结论：

$$d_M(x,h) \approx (nh)^{-1}\sigma^2 c_K + h^4 d^2{}_K [m''(x)]^2 / 4 \qquad (5)$$

这里

$$\sigma^2 = \mathrm{var}(\varepsilon_i),\ c_K = \int K^2(u)du,\ d_K = \int u^2 K(u)du$$

当 h增大时，偏差bias增加的时候方差会下降。.

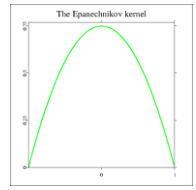


Figure 2. The Epanechnikov kernel
K (u) = 0.75(1-u^2*) I (|u| <= 1).*

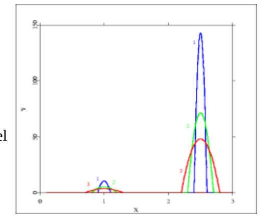
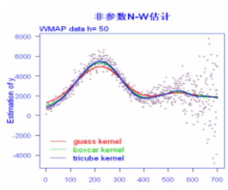


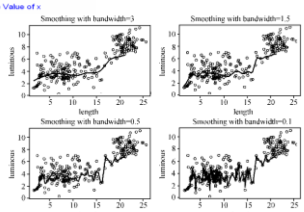
Figure 3. The effective kernel weights for the food versus net income data set. $K_h(x-\cdot)/\hat{f}_h(x)$ at x = 1 and x = 2.5 for h = 0.1 (label 1), h = 0.2 (label 2), h = 0.3 (label 3) with Epanechnikov kernel.



N-W估计中核的选择影响微乎其微，带宽的影响比较大

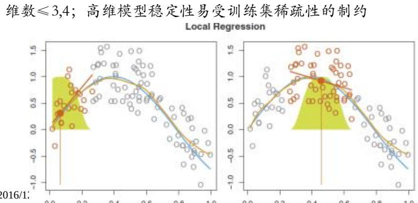
The amount of averaging is controlled by a *smoothing parameter*. The choice of smoothing parameter is related to the balances between *bias* and *variance*.

带宽变化时模式的变化



局部回归 -Local Regression

- 局部回归方法：
  - 取每个局部点x_0附近，长度$s=k/n$的邻域分段
  - 依据距离，为邻域内点赋予权重K_0，外围点权重为0
  - 最小二乘拟合，使估计参数满足：$min\sum_{i=1}^{n} K_{i0}(y_i - \beta_0 - \beta_1 x_i)^2$
  - 联合各点函数拟合预测模型
- 自变量较多，可考虑有选择的选取自变量进行局部回归
- 维数≤3,4；高维模型稳定性易受训练集稀疏性的制约



2016/1:

2.局部多项式回归

回忆标准非参数型：

$$Y_i = m(X_i) + \varepsilon_i, \quad i = 1, \Lambda\ , n \qquad (1)$$

$$m(x) = m(x_0) + m'(x_0)(x-x_0) + \frac{m''(x_0)}{2!}(x-x_0)^2 + \mathrm{L} +$$
$$\frac{m^{(p)}(x_0)}{p!}(x-x_0)^p + O\{(x-x_0)^{p+1}\}$$

在待估计点附近做局部多项式拟合：

$$\sum_{t=1}^{n}\left\{Y_t - \sum_{j=0}^{p}\beta_j (X_t - x_0)^j\right\}^2 K_h(X_t - x_0)$$

局部多项式的矩阵表示为：

$$\min_{\beta}(y - X\beta)^T W (y - X\beta)$$

令

$$\mathbf{X} = \begin{pmatrix} 1 & X_1 - x & \cdots & (X_1 - x)^p \\ \vdots & \vdots & & \vdots \\ 1 & X_n - x & \cdots & (X_n - x)^p \end{pmatrix}$$

$$\beta = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_p \end{bmatrix}_{(p+1)\times 1}, \quad \mathbf{y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}_{n \times 1}$$

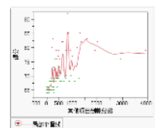
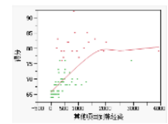
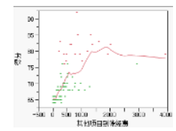
$$\mathbf{W}_{=} h^{-1}\mathrm{diag}\left[K\left(\frac{X_1 - x}{h}\right), \cdots, K\left(\frac{X_n - x}{h}\right)\right]$$

因此有加权最小二乘问题的估计 $\hat{\beta} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{y}$.

为了实现局部多项式估计，我们需要选择多项式的阶数p ，带宽h以及核函数K .当然这些参数相互关联．当 $h = \infty$ 时，局部多项式拟合就变成全局多项式拟合，阶数 p决定模型的复杂性。

与参数模型不同，局部多项式估计拟合的复杂性是由带宽来控制的，通常p是较小的，故而选择p的问题就变得不重要了．如果目的是估计m^{v}，则当$p-v$是奇数，局部多项式拟合自动修正边界偏倚．进一步，则当$p-v$是奇数，与$p-1$阶拟合相比较，p阶拟合包含了一个多余常数，但没有增加m^{v}估计的方差。不过这个参数创造了一个降低偏倚的机会，特别是在边界区域．另一方面，带宽h的选择在多项式拟合中起着重要作用．h太大的带宽引起过渡平滑，产生过大的建模偏倚，而太小的带宽会导致不足平滑，获得受干扰的估计。

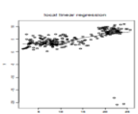
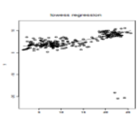
局部回归中不同的窗宽结果

14

3.稳健回归LOWESS

locally weighted scatterplot smoother

- 基本思想：

局部线性估计

稳健的权重平滑

（残差大的减小权重）

第一步：对模型 (9.2.6) 进行局部线性估计，得到 $n(X_i)$ 的估计 $\hat{m}(X_i)$，进而得到残差 $r_i = Y_i - \hat{m}(X_i)$．

第二步：计算稳健权数 $\delta_i = B(r_i/(6\cdot$ MAD $)))$，其中 $B(t) = (1-|t|^2)^2 I_{[-1,1]}(t)$．

$$I_{[-1,1]}(t) = \begin{cases} 1, & 当|t| \le 1时 \\ 0, & 当|t| > 1时 \end{cases}$$

MAD=median(|ri-median(ri)|)

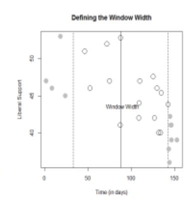
第三步：使用权 $\delta_i\, K(h_n^{-1}(X_i - x))$ 对模型 (9.0.1) 进行进行局部加权最小二乘估计，就可得到新的 r_i．

第四步：重复第二和第三步 s 次后就可得到稳健估计．

15

Step 1: Defining the window width

- The first step is to define the ***window width*** m, that encloses the closest neighbours to each data observation (the ***window half-width*** is labelled h)
  - For this example, we use m=16 (*i.e.*, for each data point we select the 16 nearest neighbours in terms of their *X*-value)
    - 16 was chosen to represent 60% of the data
    - The researcher typically chooses the window width by trial and error (more on this later)
  - The graph on the following page shows the 16 closest observations to $X_{(10)}$=88. Here we call $X_{(10)}$ our ***focal X***
- Although for this example I start at $X_{(10)}$, in the real case we would start with the first observation and move through the data, finding the 16 closest observations to each case

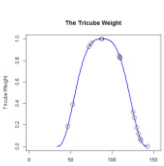



```
#Step1
#Defining the window width
plot(TIME, LIBERAL, xlab="Time (in days)", ylab="Liberal Support",
   type='n', main="Defining the Window Width")
ord <- order(TIME)
Lib=LIBERAL
time <- TIME[ord]
pre <- LIBERAL[ord]
x0 <- time[10]
```

```
diffs <- abs(time - x0)
which.diff <- sort(diffs)[16]
abline(v=c(x0-which.diff, x0+which.diff), lty=2)
abline(v=x0)
points(time[diffs > which.diff], Lib[diffs > which.diff],
pch=16, cex=2, col=gray(.75))
points(time[diffs <= which.diff], Lib[diffs <=
which.diff],cex=2)
x.n <- time[diffs <= which.diff]
y.n <- Lib[diffs <= which.diff]
text(locator(1), "Window Width")
```

Step 2: Weighting the data

- We then choose a weight function to give greatest weight to observations that are closest to the focal *X* observation
  - In practice, the ***tricube weight*** function is usually used
- Let $z_i=(x_i-x_0)/h$, which is the scaled distance between the predictor value for the *i*th observation and the focal x



$$W_T(z) = \begin{cases} \left(1-|z|^3\right)^3 & \text{for } |z| < 1 \\ 0 & \text{for } |z| \ge 1 \end{cases}$$

Here h_i is the half-width of the window centred on x_i

- Notice that observations more than *h* (the half-window or bandwidth of the local regression) away from the focal x receive a weight of 0

```
#Step 2
#Applying the Tricube Weight
#Tricube function
tricube <- function(z) {
   ifelse (abs(z) < 1, (1 - (abs(z))^3)^3, 0)
   }
#Bisquare weight
bisquare <- function(z) {
   ifelse (abs(z) < 1, (1 - (abs(z))^2)^2, 0)
   }
```

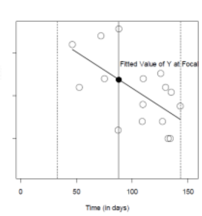
```
plot(range(TIME), c(0,1), xlab="Time (in days)",
ylab="Tricube Weight",
   type='n', main="The Tricube Weight")
abline(v=c(x0-which.diff, x0+which.diff), lty=2)
abline(v=x0)
xwts <- seq(x0-which.diff, x0+which.diff, len=250)
lines(xwts, tricube((xwts-x0)/which.diff), lty=1, lwd=2)
points(x.n, tricube((x.n - x0)/which.diff), cex=2)
```

17

- A ***polynomial regression*** using ***weighted least squares*** (using the tricube weights) is then applied to the focal X observation, using only the nearest neighbour observations to ***minimize the weighted residual sum of squares***
  - Typically a local linear regression or a local quadratic regression is used, but higher order polynomials are also possible

$$Y_i = A + B_1(x_i - x_0) + B_2(x_i - x_0)^2 + \cdots + B_p(x_i - x_0)^p + E_i$$

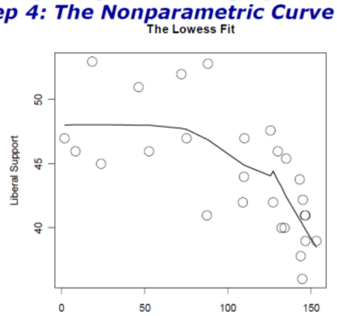
- From this regression, we then calculate the ***fitted value*** for the focal *X* value and plot it on the scatterplot
  - The regression line within the window in the following graph shows the fitted value for the focal x_i from a local linear regression




```
#Step 3
#The local polynomial
plot(TIME, LIBERAL, xlab="Time (in days)", ylab="Liberal Support",
   type='n', main="Local Linear Regression")
abline(v=c(x0-which.diff, x0+which.diff), lty=2)
abline(v=x0)
points(x.n, y.n, cex=2)
mod <- lm(y.n ~ x.n, weights=tricube((x.n-x0)/which.diff))
reg.line(mod, lwd=2, col=1)
points(x0, predict(mod, data.frame(x.n=x0)), pch=16, cex=1.8)
text(locator(1), "Fitted Value of Y at Focal X")
```

18

Step 4: The Nonparametric Curve



- Since we are trying to determine an underlying structure in the data, ***we don't want unusual cases to have extraordinary influence on the curve***
- Following from the linear regression case, M-Estimation for robust regression can be adapted to ensure that the lowess smooth is not unduly affected by outliers

1. We start by calculating the residuals from the fitted values for the local regressions

$$E_i = Y_i - \hat{Y}_i$$

2. Determine a measure of the scale of the residuals (*e.g.*, the *median absolute deviation* from the *median residual*):

$$\text{MAD} = \text{median}|E_i - \tilde{E}|$$
$$\text{where } \tilde{E} = \text{median}(E_i)$$

Adjusting for outliers (1)

3. Calculate resistance weights v_i for each observation using an appropriate weight function to determine the relative size of each residual. Here we use the ***Bisquare Weight Function***:

$$v_i \equiv w_B(z)$$
$$= \begin{cases} \left(1 - z_i^2\right)^2 & \text{for } |z| < 1 \\ 0 & \text{for } |z_i| \geq 1 \end{cases}$$

$$\text{where } z = \frac{E_i}{t \times \text{MAD}}$$
and t is a tuning constant

- t=6 MADs corresponds approximately to 4 standard deviations. In other words, we exclude observations that have a probability of being observed of less than 0.0001.

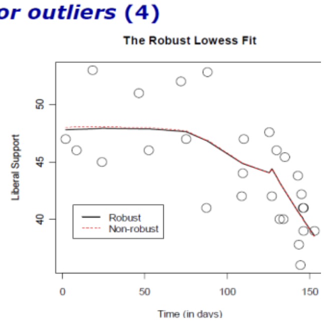
Adjusting for outliers (3)

4. We then refit the local polynomial regressions using both local weights (w_i) and the resistance weights (v_i)
5. From these new regressions, we calculate new fitted values
6. Steps 1-4 are repeated (iterated) until the fitted values stabilize
7. Finally, a curve is drawn to ***connect the fitted values***, giving us the ***lowess smooth***

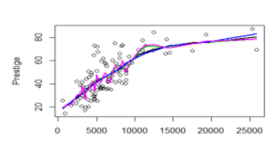
```
plot(TIME, LIBERAL, xlab="Time (in days)", ylab="Liberal Support",
    main="The Robust Lowess Fit", cex=2)
lines(lowess(TIME, LIBERAL, f=0.6), lwd=2)
lines(lowess(TIME, LIBERAL, f=0.6, iter=0),
     lty=2, col="red")
legend(locator(1), lty=c(1:2), lwd=c(2,1),
     col=c("black", "red"),
     legend=c('Robust', 'Non-robust'))
```


Adjusting for outliers (4)

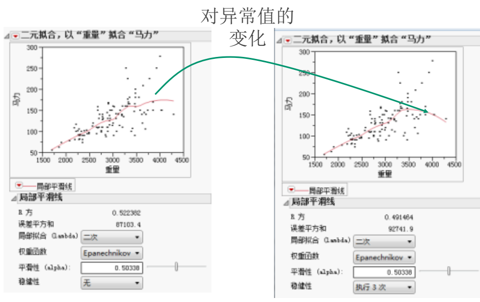
- In this case the robust fit is nearly identical to the regular lowess fit, indicating that outliers are not problematic
- Nonetheless, most lowess procedures use the robust fit by default (`locfit` is an exception)




```
library(car) # for data sets
data(Prestige)
attach(Prestige)
plot(income, prestige, xlab="Average Income", ylab="Prestige")
lines(lowess(income, prestige, f=0.5, iter=0), lwd=2)
lines(lowess(income, prestige, f=0.8, iter=0), lwd=2,col=4)
lines(lowess(income, prestige, f=0.1, iter=0), lwd=2,col=6)
```



普通的局部多项式回归　　稳健的多项式回归



Interpreting the Local Regression Estimate

- In linear regression our interest is in the regression coefficients, in particular the slopes
  - Our interest, then, is in how well the estimated coefficients represent the true population coefficients
  - We focus on confidence intervals and t-test for individual coefficients
- In nonparametric regression we have ***no parameter estimates*** (hence the name "nonparametric")
  - Our interest is on the fitted curve
  - We calculate estimates and confidence intervals (or envelopes) but they are with respect to the complete curve rather than a particular estimate
  - In other words, we focus on how well the estimated curve represents the population curve

案例:NOx排放量与发动机性能之间的关系

背景：重度雾霾政策解读----减少机动车行驶

已有的研究

•**发动机压缩比：**高压缩比发动机高温作业下，可引发轻微爆燃现象，导致NOx排放量增加。

•**燃料空气当量比：**燃料与空气比例小于1或在1附近时，对应着空气未得到完全燃烧，造成燃烧效率低下，产生较多尾气。

问题：

•两个变量对Nox实际会产生怎样的影响？

•影响的模式是怎样的？

•模型中的参数是怎样估计的？

有没有关系

关系如何定义

稳定的关系是通过参数如何控制的

Data：NOx排放物数据ethanol

排放物NOx成分多少

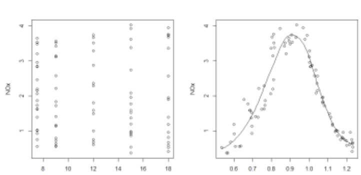
发动机的压缩比

燃料-空气当量比

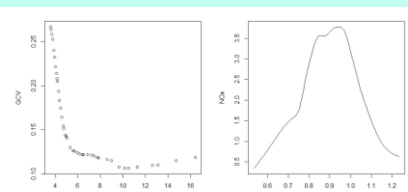
NOx	CompRatio	EquivRatio
3.741	12	0.907
2.295	12	0.761
1.498	12	1.108
2.881	12	1.016
0.76	12	1.189
3.12	9	1.001
0.638	9	1.231
1.17	9	1.123
2.358	12	1.042
0.606	12	1.215

散点图和局部线性模型

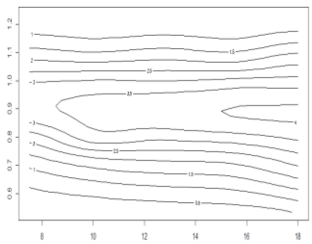
```
plot(NOx~C,data=ethanol)
fit=locfit(NOx~lp(E,nn=0.5),data=ethanol)
plot(E,NOx,data=ethanol)
lines(fit)
```




```
#cross-validation
alpha =seq(0.2,1,by=0.02)
n1=length(alpha)
g=matrix(nrow=n1,ncol=4)
for (k in 1:length(alpha))
{ g[k,]=gcv(NOx~lp(E,nn=alpha[k]),data=ethanol)}
plot(g[,4]~g[,3],ylab="GCV",xlab="degrees of freedom")
f1=locfit(NOx~lp(E,nn=0.3),data=ethanol)
plot(f1)
```



```
fit1=locfit(NOx~lp(C,E,nn=0.3,scale=0),data=ethanol)
plot(fit1)
```



高排量
汽车

发动机不
充分燃烧

31