非参数计量方法与Stata应用

王群勇（南开大学数量经济研究所所长，教授、博士生导师，QunyongWang@outlook.com）
上海财经大学；2019年8月20号

非参数模型

参数模型：

$$E(y|x) = f(x;\beta).$$

为了避免模型的错误设定，非参数方法用未知形式的函数代替。

$$E(y|x) = f(x)$$

其中，$f(x)$的形式是未知的，是需要估计的。

两大类非参数估计量：核估计（局部平滑），序列估计（全局平滑）。 非参数方法对函数形式的稳健性是有代价的，非参方法需要更多的观测值和更长的计算时间，而且计算量随着变量个数增加而指数增加(维度诅咒)。

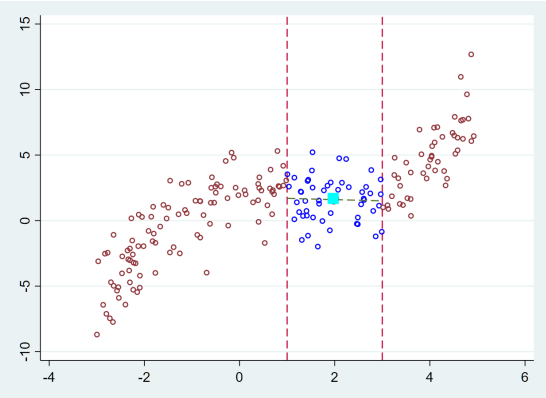
核回归

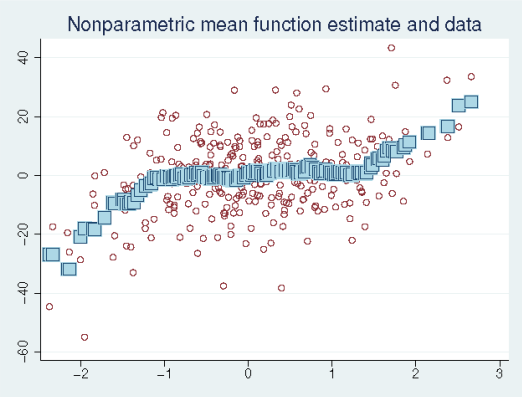
局部LS回归

非参模型

$$E(y_i|x_i = x) = m(x)$$

核回归是最小二乘法局部化，即在某个局部区间利用最小二乘法进行预测。下面的图展示了核回归的核心思想。



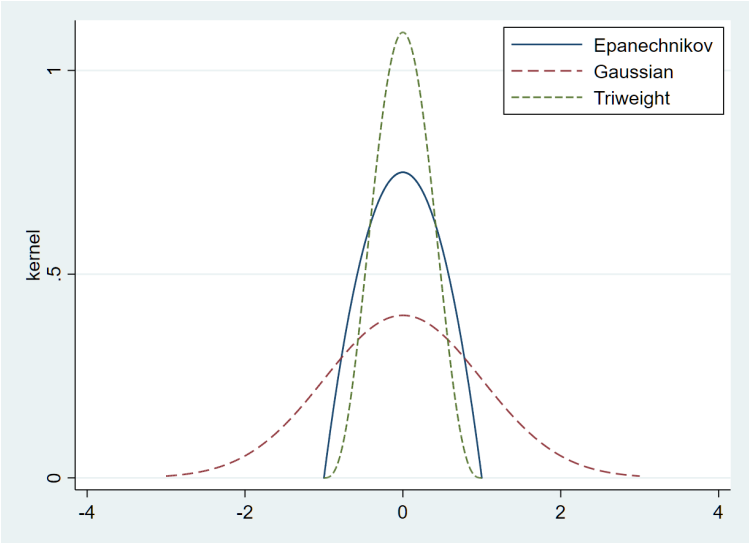


1. 不同观测值的权重如何设定 – 如何选择核函数?
2. 包括附近多少观测值–如何选择窗宽?

核函数定义：(1) $\int k(u)du = 1$; (2) $k(u) = k(-u)$; (3) $\int u^2 k(u)du > 0$.

核函数	公式
`Epanechnikov` (parabolic)	$k(u) = \frac{3}{4}(1-u^2)1(\|u\| \leq 1)$
`biweight`	$k(u) = \frac{15}{16}(1-u^2)1(\|u\| \leq 1)$
`triangle`	$k(u) = (1-\|u\|)1(\|u\| \leq 1)$
`rectangle` (uniform)	$k(u) = \frac{1}{2}1(\|u\| \leq 1)$
`cosine`	$k(u) = \frac{\pi}{4}\cos(\pi u/2)1(\|u\| \leq 1)$
`gaussian`	$k(u) = \frac{1}{\sqrt{(2\pi)}}\exp(-u^2/2)$

下图是Epanechenikov、triweight和Gaussian核的形状。



核函数

Nadaraya-Watson估计量：在某个点x上，$E(y_i|x_i=x)$的估计量为

$$\hat{m}(x)=\min\sum_{i=1}^{n}(y_i-m(x))^2K_h(x_i,x).$$

令$a=m(x)$，一阶条件为

$$-2\sum(y_i-a)K_h(x_i,x)=0$$

得到

$$\hat{m}(x)=\frac{\sum y_iK_h(x_i,x)}{\sum K_h(x_i,x)}=\sum y_iA_i.$$

这一估计量也称为局部常数（local constant）LS估计。

一般的核密度估计量为

$$\hat{f}(x)=(N)^{-1}\sum_{i=1}^{N}\frac{1}{h}k\left(\frac{x_i-x}{h}\right)$$

可以视为y_i在点x的常数近似，即

$$y_i=m(x_i)+u_i\approx m(x)+u_i.$$

也可以做线性近似（Taylor一阶展开），

$$y_i=m(x_i)+u_i\approx m(x)+(x_i-x)\beta(x)+u_i$$

令$a=m(x),b=\beta(x)$，那么

$$y_i=a+b(x_i-x)+v_i$$

局部LS估计量最小化残差平方和

$$\min\sum_{i=1}^{n}(y_i-a-(x_i-x)b)^2K_h(x_i,x).$$

这一估计量称为局部线性（local linear）LS估计。

类似地，可以做更高阶的展开，称为局部多项式LS回归。展开的阶数越高，估计的偏差越小，但波动越大。一般来讲，如果我们对p阶导数感兴趣， 那么应该做p阶展开。比如，经济学中往往关注x对y的边际影响，即一阶导数，那么 应该做一阶近似。

窗宽选择

根据理论确定最优数值，或者根据数据进行确定（交叉校正法，cross-validation）。

交互校正法

交互校正（cross-validation）方法是估计窗宽h使其最小化目标函数

$$LSCV(h) = \sum_{i=1}^{n}[y_i - \hat{m}_{-i}(x_i)]^2.$$

其中，$\hat{m}_{-i}$称为leave-one-out估计量，即把观测值i删掉后得到的估计量。

h没有明确的解析解，可以通过数值最优化方法计算。比如，Powell (2009)的BOBYQA算法和Nelder-Mead算法等，都不需要计算一阶导数。

另外一种方法是利用格点搜寻法求解h。在一组备用窗宽数值$(h_1, \ldots, h_J)$中，依次估计$LCSV(h)$， 选择使$LCSV(h)$最小的h。

信息准则

也可以信息准则（AIB, Hurvich, Simonoff, and Tsai, 1998）选择最优窗宽。

标准差和置信区间

根据线性模型的回归理论，$y = x\beta + u$在点x_0处的拟合值（即OLS预测值）为，

$$\hat{y}_0 = x_0\hat{\beta}.$$

根据线性模型的经典公式计算$\hat{y}_0$的标准差和置信区间。

另外一种方法是采用自举标准差。

配对自举（paired bootstrap）

1. 对$w_i = (x_i, y_i)$进行有放回地抽样，令(y_i^*, x_i^*)为配对自举样本。
2. 使用配对自举样本在点x上估计$m(x)$，得到$\hat{m}(x)_1^*$。注意，这里估计的点x与原始样本的估计的点必须保持一致。
3. 重复步骤(1)-(2) R次，得到$(\hat{m}(x)_1^*, \hat{m}(x)_2^*, \ldots, \hat{m}(x)_R^*)$。 进而计算$\hat{m}(x)$的自举标准差或置信区间。

残差自举（residual bootstrap）

令$\hat{u}_i = y_i - \hat{m}(x_i)$，定义$\tilde{u}_i = \hat{u}_i - \bar{\hat{u}}$。

1. 对$\tilde{u}_i$进行有放回地抽样，生成$y_i^* = \hat{m}(x_i) + \tilde{u}^*$， (y_i^*, x_i)为残差自举样本。
2. 使用残差自举样本在点x上估计$m(x)$，得到$\hat{m}(x)_1^*$。 注意，这里估计的点x与原始样本的估计的点必须保持一致。
3. 重复步骤(1)-(2) R次，得到$(\hat{m}(x)_1^*, \hat{m}(x)_2^*, \ldots, \hat{m}(x)_R^*)$。 进而计算$\hat{m}(x)$的自举标准差或置信区间。

野蛮自举（wild bootstrap）

令$\hat{u}_i = y_i - \hat{m}(x_i)$，定义$\tilde{u}_i = \hat{u}_i - \bar{\hat{u}}$。

1. 对$\tilde{u}_i$进行有放回地抽样（Mammen, 1993），

$$\check{u}_i^* = \begin{cases} \tilde{u}_i(1 - \sqrt{5}/2 \approx -0.6180\tilde{u}, & \text{with prob. } (1+\sqrt{5})/(2\sqrt{5}) \approx 0.7236 \\ \tilde{u}_i(1 + \sqrt{5}/2 \approx 1.6180\tilde{u}, & \text{with prob. } (\sqrt{5}-1)/(2\sqrt{5}) \approx 0.2764 \end{cases}$$

生成$y_i^* = \hat{m}(x_i) + \check{u}^*$， (y_i^*, x_i)为野蛮自举样本。

或者按照Rademacher分布抽样，

$$\check{u}_i^* = \begin{cases} -\tilde{u}_i, & \text{with prob. } 0.5 \\ \tilde{u}_i, & \text{with prob. } 0.5 \end{cases}$$

2. 使用野蛮自举样本在点x上估计$m(x)$，得到$\hat{m}(x)_1^*$。

注意，这里估计的点x与原始样本的估计的点必须保持一致。

3. 重复步骤(1)-(2) R次，得到$(\hat{m}(x)_1^*, \hat{m}(x)_2^*, \ldots, \hat{m}(x)_R^*)$。 进而计算$\hat{m}(x)$的自举标准差或置信区间。

自举参数没有绝对的标准。可以尝试不同的自举次数，观察不同自举次数得到的自举标准差的变化是否比较大。如果变化比较大，意味着标准差不够稳定，应该增加自举次数。

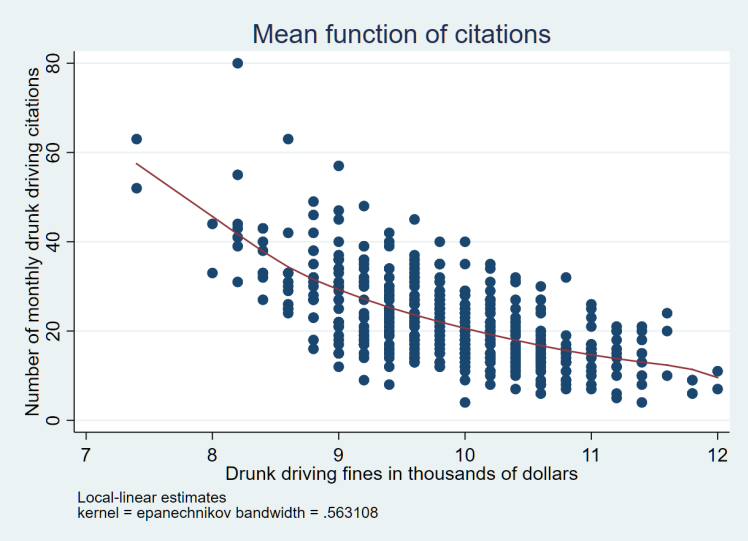
应用案例

核回归的Stata指令为：

```
npregress kernel [depvar] [indep], estimator() kernel() dkernel() predict() vce() reps() imaic
```

选项	说明	例子
estimator()	constant, linear(默认选项)	estimator(linear)
kernel()	连续变量的核函数	kernel(gaussian)
dkernel()	离散变量的核函数，包括: liracine, cellmean	dkernel(liracin)
predict()	均值和导数的新变量名称	predict(mean deriv, replace)
vce()	协方差计算方法，包括：none, boostrap	vce(none)
reps()	自举次数	reps(200)
imaic	修正的AIC	imaic

```
. use dui, clear
(Fictional data on monthly drunk driving citations)

. npregress kernel citations fines

Computing mean function

Minimizing cross-validation function:

Iteration 0:   Cross-validation criterion =  35.478784
Iteration 1:   Cross-validation criterion =  4.0147129
Iteration 2:   Cross-validation criterion =  4.0104176
Iteration 3:   Cross-validation criterion =  4.0104176
Iteration 4:   Cross-validation criterion =  4.0104176
Iteration 5:   Cross-validation criterion =  4.0104176
Iteration 6:   Cross-validation criterion =  4.0104006

Computing optimal derivative bandwidth

Iteration 0:   Cross-validation criterion =  6.1648059
Iteration 1:   Cross-validation criterion =  4.3597488
Iteration 2:   Cross-validation criterion =  4.3597488
Iteration 3:   Cross-validation criterion =  4.3597488
Iteration 4:   Cross-validation criterion =  4.3597488
Iteration 5:   Cross-validation criterion =  4.3597488
Iteration 6:   Cross-validation criterion =  4.3595842
Iteration 7:   Cross-validation criterion =  4.3594713
Iteration 8:   Cross-validation criterion =  4.3594713

Bandwidth
------------------------------------------
             |      Mean     Effect
-------------+----------------------------
       fines |   .5631079    .924924
------------------------------------------

Local-linear regression                   Number of obs      =            500
Kernel   : epanechnikov                   E(Kernel obs)      =            282
Bandwidth: cross validation               R-squared          =         0.4380
------------------------------------------------------------------------------
   citations |   Estimate
-------------+----------------------------------------------------------------
Mean
   citations |   22.33999
-------------+----------------------------------------------------------------
Effect
       fines |  -7.692388
------------------------------------------------------------------------------
Note: Effect estimates are averages of derivatives.
Note: You may compute standard errors using vce(bootstrap) or reps().

. npgraph
```



核回归均值估计

非参数模型体现的是变量x和y的非线性关系，和x对y的非线性影响。在x的不同点上， Stata给出了期望值$E(y|x)$的估计量$\hat{E}(y|x)$，及其一阶导数$\partial E(y|x)/\partial x$的估计量， 即$\partial \hat{E}(y|x)\partial x$。Stata自动生成两个系统变量，**_Mean_dep**和**_d_Mean_dep**。

```
des *_*, fullnames
```

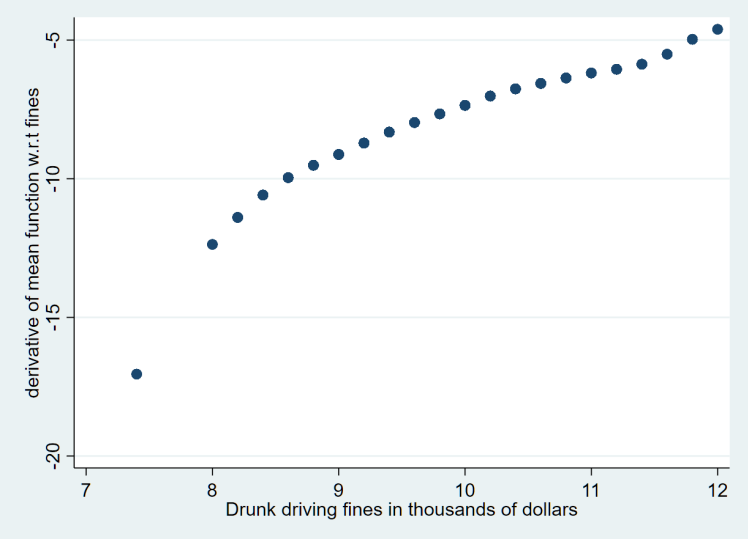
这两个系统变量也可以通过npregress的选项predict指定名称。

```
npregress kernel citations fines, predict(mean deriv) nolog
```

或者在npregress之后，通过predict预测生成。

```
predict deriv, deriv
summ deriv
scatter deriv fines
```

下图是边际效应图。



边际效应

Stata默认不计算估计量的标准差。要计算标准差和置信区间，设定reps()自举次数。要得到可重复的结果，则通过seed()设定随机数生成的种子值。

```
npregress kernel citations fines, reps(50) seed(123)
npregress kernel citations fines, reps(50) seed(123) predict(mean deriv, replace)
summ mean deriv
```

在模型中加入其它控制变量。连续变量和分类变量采用不同的核函数，因此对于分类变量，必须采用Stata的因子符号i.来表示。

```
npregress kernel citations fines i.taxes i.csize i.college, nolog reps(100) seed(12)
predict deriv*, deriv
```

```
summ deriv*
```

利用margins计算不同情境下的效果比较。 比如，如果罚金提高15%，那么酒驾的水平是多少?

```
. margins, at(fines=generate(fines*1.15)) reps(50) seed(12)
(running margins on estimation sample)

Bootstrap replications (50)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50

Adjusted predictions                            Number of obs     =        500
                                                Replications      =         50

Expression   : mean function, predict()
at           : fines           = fines*1.15

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Percentile
             |     Margin   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _cons |   12.60535   .7873202    16.01   0.000     10.67344    14.09854
------------------------------------------------------------------------------
```

或者将罚金提高15%，那么酒驾的变化是多少?

```
. margins, at(fines=generate(fines)) at(fines=generate(fines*1.15)) contrast(atcontrast(r) nowald) reps(50) seed(12)
(running margins on estimation sample)

Bootstrap replications (50)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
Contrasts of predictive margins
                                                Number of obs     =        500
                                                Replications      =         50

Expression   : mean function, predict()

1._at        : fines           = fines

2._at        : fines           = fines*1.15

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Percentile
             |   Contrast   Std. Err.      [95% Conf. Interval]
-------------+----------------------------------------------------------------
         _at |
   (2 vs 1)  |  -9.734647   .7014967    -11.05981   -7.979817
------------------------------------------------------------------------------
```

或者令罚金从10增加到11，那么酒驾的新的水平：

```
margins, at(fines=10) at(fines=11) reps(50) seed(12)
```

令罚金从10增加到11，那么酒驾的变化数量及其显著性检验：

```
margins, at(fines=10) at(fines=11) contrast(atcontrast(r) nowald) reps(50) seed(12)
```

或者其他变量固定在某些特定情形，令罚金从10增加到11，那么酒驾的变化情况：

```
margins, at(fines=10 taxes=1 csize=2 college=1) ///
 at(fines=11 taxes=1 csize=2 college=1) reps(50) seed(12)
margins, at(fines=10 taxes=1 csize=2 college=1) ///
 at(fines=11 taxes=1 csize=2 college=1) ///
 contrast(atcontrast(r) nowald) reps(50) seed(12)
```

或者其他变量固定在某些特定情形，令罚金从8逐步增加到12，那么酒驾会怎么变化?

```
margins, at(fines=(8(0.5)12) taxes=1 csize=2 college=1) reps(50) seed(12)
marginsplot
```

```
margins, at(fines=(8(0.5)12) taxes=1 csize=2 college=1) contrast(atcontrast(ar)) reps(50) seed(12)
marginsplot, yline(0)
```

相比较线性模型的$R^2 = 0.72$，非参模型的R^2提高到0.8。一般而言，非参模型比参数模型可以更好地拟合数据，但计算更耗时。

模型设定检验

检验参数模型（线性或非线性）是否恰当。以线性模型为例。

原假设：$E(y|x) = x\beta$。备择假设：$E(y|x) = m(x)$.

$$SSR_p = \sum_{i=1}^{n}(y_i - x\hat{\beta})^2$$

$$SSR_{np} = \sum_{i=1}^{n}(y_i - \hat{m}(x_i))^2$$

构建统计量

$$J_n = \frac{SSR_p - SSR_{np}}{SSR_{np}}$$

J_n的p值通过野蛮自举法来计算。

1. 用原始样本计算J_n。
2. 用自举样本计算J_n^*。
3. 重复(2) R次，$p = R^{-1}\sum_{j=1}^{R} 1(J_n^* > J_n)$。

下面通过自举法对模型的设定进行检验。

原假设：线性模型（参数模型）。备择假设：非参数模型。

程序npbstest.ado计算线性模型的残差平方和以及非参回归的残差平方和，以及二者之间的差。 下面的指令在Stata中进行自举检验，自举次数为50，自举的结果保存在npbs.dta文件中。

```
. set seed 123456

. bootstrap F = r(F), reps(300) saving(npbs, replace): npbstest citation fines i.taxes i.csize
i.college
(running npbstest on estimation sample)

Warning:  Because npbstest is not an estimation command or does not set e(sample), bootstrap has no
way to determine which observations are used in calculating the
          statistics and so assumes that all observations are used.  This means that no observations
will be excluded from the resampling because of missing values or
          other reasons.

          If the assumption is not true, press Break, save the data, and drop the observations that
are to be excluded.  Be sure that the dataset in memory contains only
          the relevant data.

Bootstrap replications (300)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
..................................................   150
..................................................   200
..................................................   250
..................................................   300

Bootstrap results                                 Number of obs     =        500
                                                  Replications      =        300

      command:  npbstest citation fines i.taxes i.csize i.college
            F:  r(F)

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           F |   208.9938   82.47544     2.53   0.011     47.34487    370.6427
------------------------------------------------------------------------------
```

样本的残差平方和差为3987.7，自举标准差为1302.8。但输出的z统计量对应的原假设为系数为0， 因此，其z统计量和p值都不是我们需要的。打开所保存的自举文件npbs，然后计算J_n统计量 大于真实统计量的概率，即$P(J_n > 3987.7)$。

```
. use npbs, clear
(bootstrap: npbstest)

. count if F>208.994
  230

. dis r(N)/_N
.76666667
```

自举程序npbstest.ado的代码如下。

```
capture prog drop npbstest
program npbstest, rclass
syntax varlist(ts fv) [if] [in]
marksample touse

qui reg `varlist' if `touse'
local rssp = e(rss)
ereturn clear

qui npregress kernel `varlist' if `touse'
tempvar res
qui predict `res' if e(sample), residual
qui summ `res'
local sigsq = r(Var)
local rssnp = r(Var)*(r(N)-1)
ereturn clear
return scalar F = (`rssp'-`rssnp')/`sigsq'
end
```

全局非参估计

序列估计与样条函数

序列方法（series method）方法是利用协变量的函数来近似未知函数，这种函数称之为基础函数（basis function)。基础函数是通过多个函数来近似未知函数，常见的基础函数包括多项式函数、样条函数和B-样条函数等。

设y的均值函数为

$$E(y_i|x_i) = m(x_i).$$

序列估计量的均值函数估计为

$$\hat{E}(y_i|x_i) = z(x_i)\hat{\beta}.$$

其中，$z(x_i)$为解释变量的q阶基础函数（多项式函数、样条函数和B-样条函数），令$Z_i = z(x_i)$。序列估计量即OLS估计量

$$\hat{\beta} = (Z'Z)(Z'y).$$

多项式函数是常见的一种基础函数，对于$x = (x_1, \ldots, x_k)$，一阶多项式为

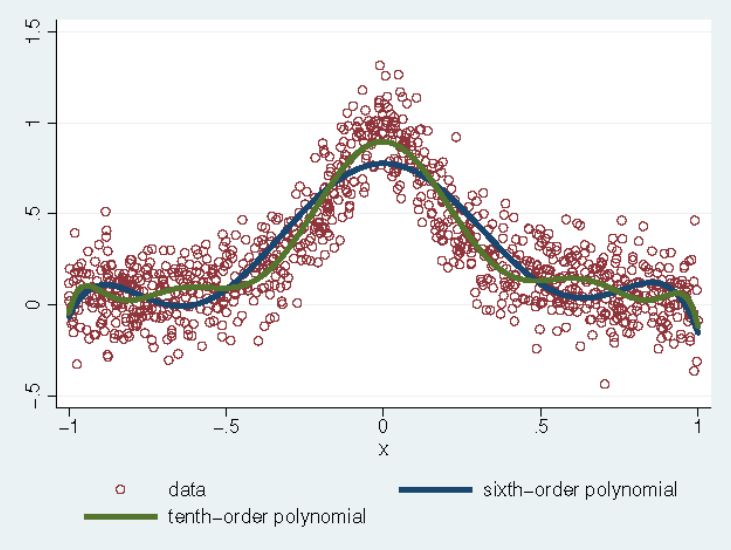
$$z(x) = (x_1, \ldots, x_k)$$

二阶多项式包括一阶多项式和所有的二次项，

$$z(x) = (x_1, \ldots, x_k; x_1^2, \ldots, x_k^2; x_1x_2, \ldots, x_{k-1}x_k).$$

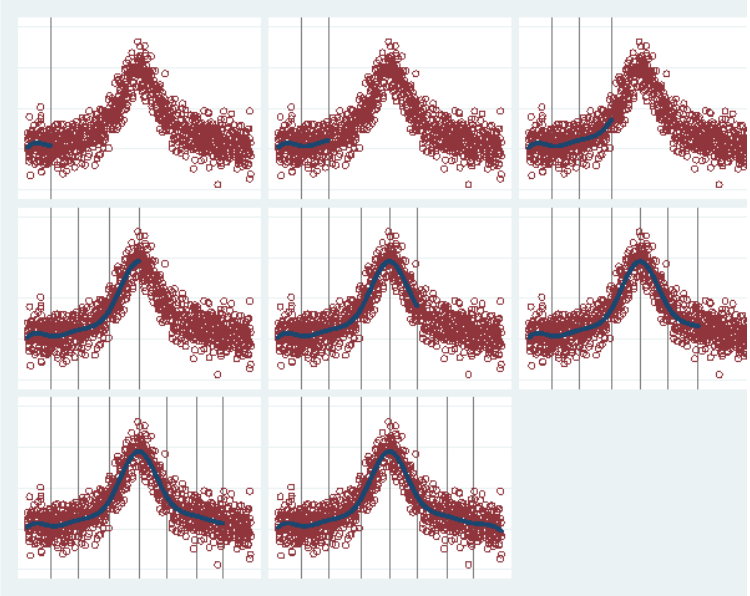
依次类推，三阶多项式包括一次项、二次项和所有的三次项。

但多项式函数存在Runge现象，即提高多项式的阶数并不能提高数据的拟合优度。



Runge现象

解决这一问题的比较好的方法是利用样条函数，即分段连续多项式函数。回归样条函数综合了多项式函数和阶跃函数， 即在不同的区间用不同的多项式来拟合，但不同区间的多项式函数式是连续的。不同区间的边界叫做结点（knot）。 样条函数通过把一组低阶多项式连接起来得到更好的拟合。



样条函数序列估计

在上面的图中，x的定义域被分为多个子区间，各子区间的边界叫作结点（knot）。每个区间采用不同的低阶多项式拟合，不同区间的多项式在结点处是连续的。在本例中，七个结点把整个区间分为八个区间，设结点为（$t_1 < t_2 < ... < t_7$）。

从这一角度来说，样条函数虽然经常被称作全局估计，但本质上也是局部估计，因为样条函数也是对不同的子区间进行估计，每个区间需要有充分多的观测值。

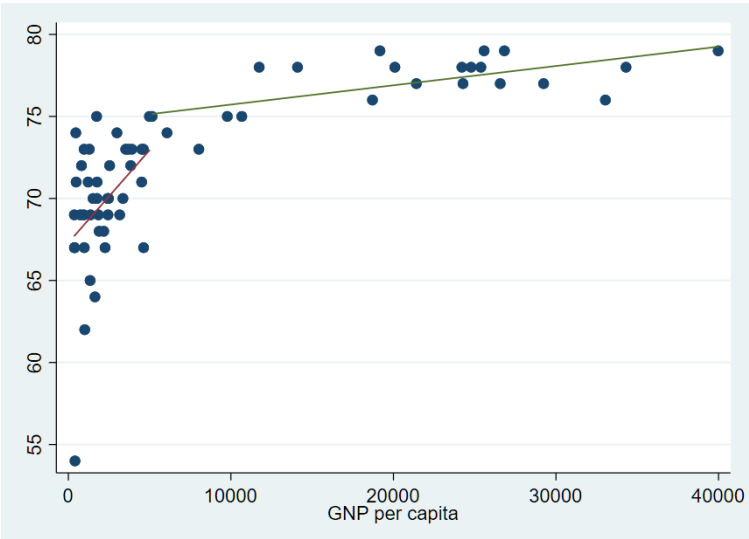
一个问题是，如何使得不同区间的多项式在结点处连续？ 更进一步，为了使得拟合曲线更平滑，如何使得不同区间的多项式在结点处连续且可导？

比如，如下分段多项式

$$y = \begin{cases} \alpha_0 + \alpha_1 x_i + \alpha_2 x_i^2 + \alpha_3 x_i^3 + \epsilon_i, & x_i < c \\ \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \epsilon_i, & x_i \geq c \end{cases}$$

上面的分段多项式是不连续的。比如，用期望寿命对人均GNP进行分段回归。

```
sysuse lifeexp, clear
twoway (scatter lexp gnppc) (lfit lexp gnppc if gnppc<5000) (lfit lexp gnppc if gnppc>=5000) ,
legend(off)
```



分段线性回归

样条函数

回归样条函数需要约束：（1）上面的分段多项式函数是连续的；（2）并且其一阶导数是连续的、二阶导数也是连续的。满足这一条件的样条函数称为三次样条（cubic spline）。

首先以分段线性函数为例。

$$m(x) = \begin{cases} \alpha_0 + \alpha_1(x-t), & x < t \\ \beta_0 + \beta_1(x-t), & x \geq t \end{cases}$$

当且仅当$\alpha_0 = \beta_0$时，$m(x)$是连续的，这时模型可以写为：

$$m(x) = \beta_0 + \beta_1 x + \beta_2(x-t)1(x \geq t).$$

设结点为t，样条函数定义为：

$$\begin{aligned} V_1 &= \min(x, t) \\ V_2 &= \max(x, t) - t \end{aligned}$$

此时，具有连续回归线的模型为

$$y = \beta_0 + \beta_1 \min(x,t) + \beta_2(\max(x,t) - t) + u = \begin{cases} \beta_0 + \beta_1 x & x < t \\ \beta_0 + \beta_1 t + \beta_2(x-t) & x \geq t \end{cases}$$

给定多个结点$(t_1, \ldots, t_K)$，线性样条函数的一般定义：

$$\begin{aligned} V_1 &= \min(x, t_1) \\ V_i &= \max(\min(x, t_i), t_{i-1}) - t_{i-1}, i = 2, \ldots, K \\ V_{K+1} &= \max(x, t_K) - t_K \end{aligned}$$

给定已知的结点(#1, #2, ...)，Stata生成线性样条函数的指令为：

`mkspline` *newvar-1* #1 [*newvar-2* #2 ...] *newvar-k* = *oldvar* [*if*] [*in*] , [`displayknots`]

按照等区间方法，Stata生成线性样条函数的指令为：

`mkspline` *stubname* # = *oldvar* [*if*] [*in*] , [`displayknots`]

按照分位数方法，Stata生成线性样条函数的指令为：

`mkspline` *stubname* # = *oldvar* [*if*] [*in*] , [`pctile displayknots`]

比如，人均期望寿命的例子。为了得到连续的拟合函数，我们生成样条函数。

```
mkspline x1 5000 x2 = gnppc  // spline function given one knot
reg lexp x1 x2
predict yf
twoway (scatter lexp gnppc) (line yf gnppc , sort(gnppc))
```

再以二次多项式函数为例。

$$m(x) = \begin{cases} \alpha_0 + \alpha_1(x-t) + \alpha_2(x-t)^2, & x < t \\ \beta_0 + \beta_1(x-t) + \beta_2(x-t)^2, & x \geq t \end{cases}$$

当且仅当$\alpha_0 = \beta_0$时，$m(x)$在$x = t$处是连续的，当$\alpha_1 = \beta_1$时，模型的一阶导数是连续的。 这时模型可以写为：

$$m(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3(x-t)^2 1(x \geq t).$$

类似地，具有连续的一阶导数和二阶导数的分段三次函数可以写为

$$m(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4(x-t)^3 1(x \geq t).$$

一般地，具有K个结点$(t_1, t_2, \ldots, t_K)$的p阶样条函数为

$$m(x) = \sum_{j=0}^{p} \beta_j x^j + \sum_{k=1}^{K} \gamma_k (x - t_k)^p 1(x \geq t_k).$$

经验应用中，常常给定p，选择结点个数K。结点$(t_1, t_2, \ldots, t_K)$可以选择为x的K个分位数， 或者将x均匀地分为$(K+1)$个区间。

自然样条函数

自然样条函数（natural spline）是约束回归样条函数在第一个结点之前为线性，在最后一个结点之后为线性， 而中间的区间为三次多项式。自然样条函数可以得到端点的更稳定的估计。 自然样条函数有时也叫做受约束的三次样条函数（restricted cubic spline）。

Stata生成自然样条函数的指令为：

mkspline *stubname* = *oldvar* [*if*] [*in*] [*weight*] , cubic [nknots(#) knots(*numlist*) displayknots]

结点的个数至少为2个。可以通过knots()自己设置结点，或者通过nknots()设定结点的个数， Stata自动根据Harrell (2001, 23)按照分位数来设置结点，具体如下。

- K=3: 10, 50, 90
- K=4: 5, 35, 65, 95
- K=5: 5, 27.5, 50, 72.5, 95
- K=6: 5, 23, 41, 59, 77, 95
- K=7: 2.5, 18.33, 34.17, 50, 65.83, 81.67, 97.5

如果通过nknots()设定结点，结点个数必须介于[3, 7]之间。

以酒驾数据为例。

```
use dui, clear
mkspline spfines 5 = fines, pctile displayknots
des spfines*
reg citations spfines1-spfines5 i.taxes i.csize i.college

label var spfines1 "fines: (0,9.2)"
label var spfines2 "fines: (9.2,9.6)"
label var spfines3 "fines: (9.6,10.2)"
label var spfines4 "fines: (10.2,10.6)"
label var spfines5 "fines: (10.6,.)"
coefplot , keep(sp*)
```

三阶样条函数定义为

$$E(y_i|x_i) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \sum_{j=1}^{7} \beta_{j+3} \max(x_i - t_j, 0)^3 = \begin{cases} \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3, & x_i \le t_1 \\ \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 (x_i - t_1)^3, & t_1 < x_i \le t_2 \\ \dots & \dots \\ \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_7 (x_i - t_7)^3, & t_7 < x_i \end{cases}$$

B-样条函数

自然样条函数存在高度共线性问题，B-样条函数则通过令不同区间的基础函数正交克服了这一缺点。 B样条函数由de Boor(1978)和Ziegler(1969)提出。

在自然样条函数和B-样条函数中，连续变量需重新标度为介于[0,1]区间的变量。

$$(x_i - x_{\min}) \frac{1}{x_{\max} - x_{\min}}$$

对于多个解释变量，基础函数（basis function）包括每个变量的多项式及其交叉积。比如，两个变量$((x, z))$，那么基础函数包括$(x, z, xz, x^2, z^2, x^2z^2, \dots, x^k, z^k, x^kz^k)$。

相当于设定模型的形式为

$$m(x, z) = g_1(x) + g_2(z).$$

Stata可以通过nointeract选项设定方程中没有交叉项。

如果某些变量以线性形式加入到模型中，那么Stata中设定asis()选项。

```
npregress series, options
```

选项	注释
bspline或者bspline(#)	B-样条函数
spline或者spline(#)	自然样条函数
polynomial或者polynomial(#)	多项式函数
asis(*varlist*)	线性变量，即不做任何变换
nointeract(*varlist*)	不做交叉变量
criterion()	阶数的选择标准，包括cv、gcv、aic、bic和mallows
knots(#)	样条函数或B-样条函数中结点的个数

```
. use dui, clear
(Fictional data on monthly drunk driving citations)

. npregress series citations fines

Computing approximating function
```

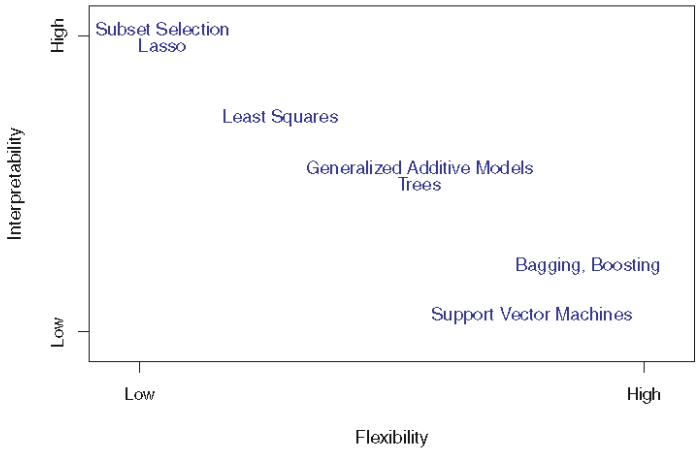
```
Minimizing cross-validation criterion

Iteration 0:  Cross-validation criterion =  55.15697
Iteration 1:  Cross-validation criterion =  55.11413

Computing average derivatives

Cubic B-spline estimation                 Number of obs      =            500
Criterion: cross validation               Number of knots    =              3
------------------------------------------------------------------------------
             |                 Robust
   citations |     Effect   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       fines |  -8.020769    .464836   -17.26   0.000    -8.931831   -7.109707
------------------------------------------------------------------------------
Note: Effect estimates are averages of derivatives.
```

`npregress series`自动生成基础函数的系统变量。

更多例子

```
use vote, clear
global xs "female edu lninc married urban han i.prov"
qui probit vote cult age $xs , vce(robust)
est store probit1
qui probit vote cult age c.age#c.age c.age#c.age#c.age $xs , vce(robust)
est store probit2
mkspline spage 3 = age, pctile displayknots
qui probit vote cult spage* $xs , vce(robust)
est store probit3
mkspline age1 20 age2 30 age3 40 age5 60 age6 = age, displayknots
qui probit vote cult age? $xs , vce(robust)
est store probit4
est table probit1 probit2 probit3 probit4, star(.1 .05 .01) eq(1)  ///
 keep(cult female edu lninc married urban han)  stat(N aic bic)
```

模型的弹性与可释性方面是相互替代的。



tradeoff between intepretability and flexibility