第二届Stata中国用户大会

大数据、高维回归与Stata

陈强
山东大学经济学院
qiang2chen2@126.com
公众号/网站：econometrics-stata
腾讯课堂：https://metrics.ke.qq.com

本讲内容

- 高维数据
- 岭回归 (Ridge Regression)
- 套索估计 (Lasso)
- 弹性网 (Elastic Net)
- Stata命令与案例

高维数据

- 大数据的一种表现形式为“高维数据”（high dimensional data），即变量个数（*p*）大于样本容量（*n*），也称为“data-rich environment”。
- 比如，某些研究收集了100位病人的信息，其中每位病人均有2万条基因的数据。受成本限制，样本容量*n*=100很难再扩大，而变量个数*p*远远大于样本容量。
- 如此之多的变量自然提供了更多的信息，但同时也为回归估计带来了新的挑战。

经济学有高维数据吗

- 经济学不仅有高维数据，而且越来越多。

- **情形一**、数据本身可能就是高维的。比如，人口普查、工业调查或家庭调查数据，通常会包括每位个体的数百个变量。

- 而交易层面的数据（包括网购与零售扫描数据）、社交媒体的数据、以及文本挖掘的数据，其变量个数则一般成千上万，甚至更多。

经济学有高维数据吗（续）

- **情形二**、尽管原始变量不多，但我们通常不知道这些变量应该以怎样的函数形式 (functional form) 进入回归方程

- 为了解决潜在非线性，可能加入原始变量的平方项、交互项、甚至更高次项，以及其他变换（比如取对数），使得最终变量个数大大增加

- 情形二在计量经济学的实证研究中一直存在

高维回归的挑战

- 高维回归的最大挑战是很容易出现“过拟合”（overfit）
- 对于 *p>n* 的高维数据，可用来解释 *y* 的 **x** 很多
- 如果使用传统的OLS回归，虽可得到完美的样本内拟合（in-sample fit），但外推预测的效果则可能很差

一个启发性例子

- 假设 $n=p=100$。即使这100个解释变量$\boldsymbol{x}$与被解释变量y毫无关系（比如，相互独立），但将y对$\boldsymbol{x}$作OLS回归，也能得到拟合优度$R^2=1$的完美拟合。由样本估计的回归函数，将毫无外推预测的价值

- 这种拟合显然过度了（故名“过拟合”），因为它不仅拟合了数据中的信号（signal），而且拟合了数据中的很多噪音（noise）。在此极端例子中，由于数据中全是噪音而毫无信号，故OLS完美地拟合了数据中的噪音，自然毫无意义。

严格多重共线性是常态

- 在$p<n$的传统计量经济学中，严格多重共线性（strict multicollinearity）比较少见；即使出现，只要将多余的变量去掉就行
- 在$p>n$的高维数据中，严格多重共线性成为常态。比如，任意$n+1$个变量之间，一般就存在严格多重共线性。
- 简单去掉导致多重共线性的变量将无济于事，因为需要扔掉很多变量（想想100个病人，2万个基因变量的例子），难免将婴儿与洗澡水一起倒掉

严格多重共线性下的OLS

- 对于*p*>*n*的高维数据，OLS一般没有唯一解
- 任意线性无关的*n*个变量 $\boldsymbol{x}$，均可完美地解释 y。
- 此时，可将OLS估计量的方差视为无穷大，因为OLS估计量的方差 $\sigma^2(\mathbf{X}'\mathbf{X})^{-1}$，而在严格多重共线性的情况下，$(\mathbf{X}'\mathbf{X})^{-1}$ 并不存在

岭回归的起源

- 作为高维回归的方法之一，岭回归（ridge regression）最早由Hoerl and Kennard (1970)提出，其出发点正是为了解决多重共线性
- 在传统的低维回归(low-dimensional regression)，虽然严格多重共线性很少见，但不完全的多重共线性却不时出现，其解释变量$\boldsymbol{x}$之间高度相关
- 矩阵 $(\mathbf{X}'\mathbf{X})$ 变得几乎不可逆，导致OLS估计量的方差 $\sigma^2(\mathbf{X}'\mathbf{X})^{-1}$ 变得很大

岭回归的解决方法

- 在矩阵 $(\mathbf{X}'\mathbf{X})$ 的主对角线上都加上常数 λ，使所得矩阵 $(\mathbf{X}'\mathbf{X}+\lambda\mathbf{I})$ 变得“正常”
- OLS估计量 $\hat{\boldsymbol{\beta}}_{OLS} \equiv (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$，而岭回归估计量为

$$\hat{\boldsymbol{\beta}}_{ridge} \equiv (\mathbf{X}'\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}$$

- 岭回归只是在OLS表达式中加了“山岭” $\lambda\mathbf{I}$，故名“岭回归”

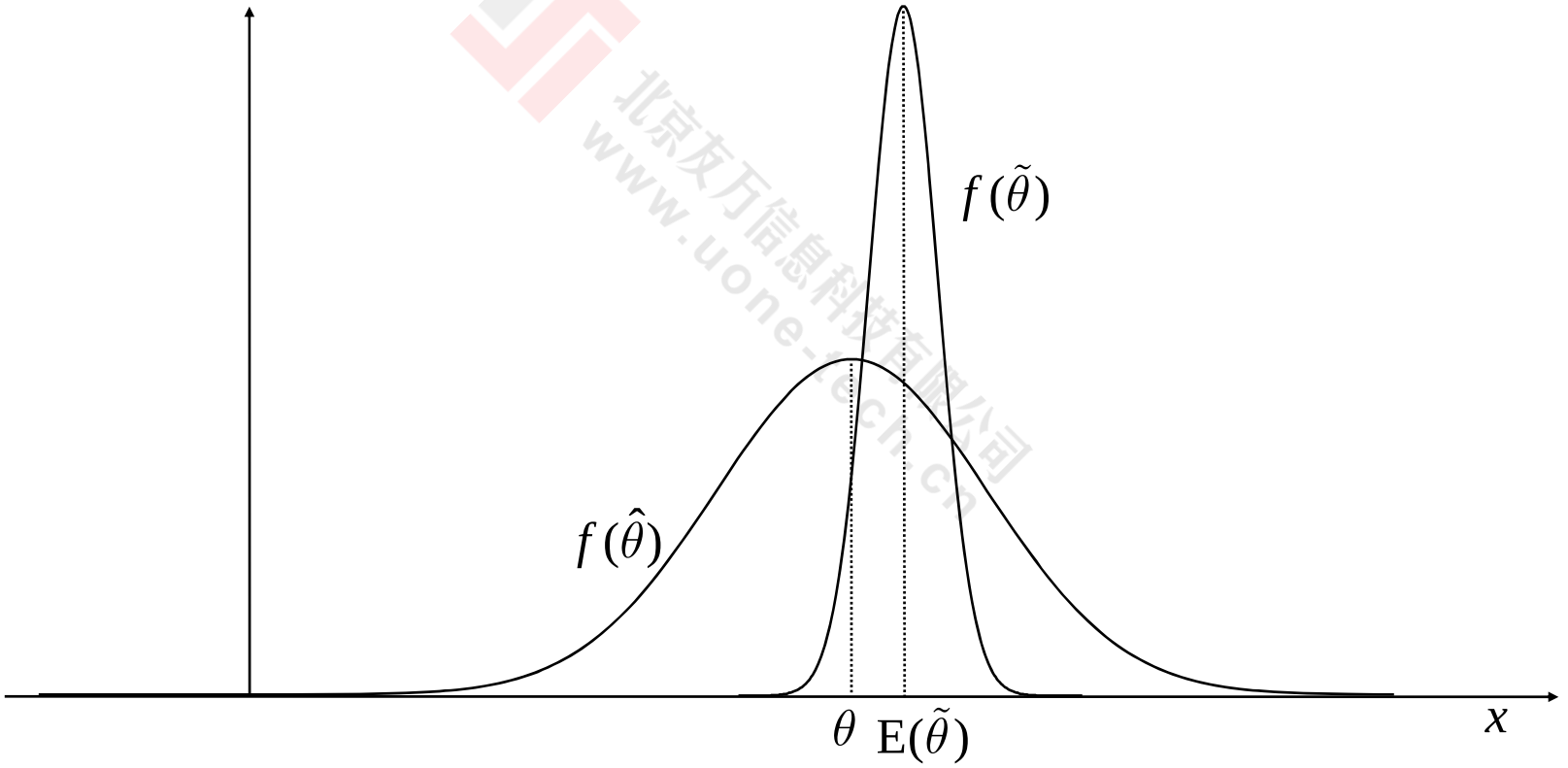
岭回归的优点

- 由于OLS估计量是无偏（unbiased），故凭空加上此“山岭”之后，所得的岭回归估计量其实是有偏的（biased）。

- 但在多重共线性的情况下，OLS估计量的方差太大，而岭回归则可减小方差，使得岭回归估计量的均方误差（MSE）可能更小（均方误差等于方差加上偏差平方）。

Bias-Variance Trade-off

$$\mathrm{MSE} = \mathrm{Bias}^2 + \mathrm{Variance}$$



惩罚回归 (Penalized Regression)

- 可将岭回归估计量看成以下最小化问题的最优解，其目标函数为残差平方和(SSR)，再加上一个惩罚项(惩罚太大的参数向量)：

$$\hat{\boldsymbol{\beta}}_{ridge} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \underbrace{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})}_{SSR} + \underbrace{\lambda \|\boldsymbol{\beta}\|_2^2}_{penalty}$$

- $\lambda > 0$ 为“微调参数”(tuning parameter)，控制惩罚的力度。$\|\boldsymbol{\beta}\|_2 = \sqrt{\beta_1^2 + \cdots + \beta_p^2}$ 为参数向量$\boldsymbol{\beta}$ 的2-范数（L_2 norm），即该向量的长度。

一阶条件

$$\min_{\boldsymbol{\beta}} \; L(\boldsymbol{\beta}) \equiv (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}'\boldsymbol{\beta}$$

$$\frac{\partial L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \equiv -2\mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + 2\lambda\boldsymbol{\beta} = \mathbf{0}$$

$$\mathbf{X}'\mathbf{X}\boldsymbol{\beta} + \lambda\boldsymbol{\beta} = \mathbf{X}'\mathbf{y}$$

$$(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})\boldsymbol{\beta} = \mathbf{X}'\mathbf{y}$$

$$\hat{\boldsymbol{\beta}}_{ridge} \equiv (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}$$

技术细节

- 变量单位对OLS回归系数没有实质影响
- 由于岭回归惩罚过大的回归系数，故变量单位对岭回归有实质影响
- 一般不希望惩罚常数项，因为常数项仅代表被解释变量的平均值

解决方法

- 在做岭回归之前，先将每个解释变量 $\boldsymbol{x}$ 都标准化，即减去其均值，除以标准差
- 不惩罚常数项，则常数项的估计值为 $\hat{\alpha} = \bar{y}$ (参见下页)
- 将被解释变量 y 中心化，使得 $\bar{y} = 0$ ，则不需要在惩罚回归中加入常数项

Details

Model: $$y_i = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i$$

FOC: $$\frac{\partial \sum_{i=1}^{n} e_i^2}{\partial \alpha} = \frac{\partial \sum_{i=1}^{n} (y_i - \alpha - \beta_1 x_{i1} - \beta_2 x_{i2})^2}{\partial \alpha}$$

$$= -2\sum_{i=1}^{n} (y_i - \alpha - \beta_1 x_{i1} - \beta_2 x_{i2}) = 0$$

$$\Rightarrow \quad \bar{y} = \hat{\alpha} + \hat{\beta}_1 \underbrace{\bar{x}_1}_{=0} + \hat{\beta}_2 \underbrace{\bar{x}_2}_{=0} = \hat{\alpha}$$

收缩估计量

- 由于目标函数中包含对过大参数的惩罚项，故岭回归为“收缩估计量”(shrinkage estimator)。
- 与OLS估计量相比，岭回归估计量更为向原点收缩
- 这可从几何上得到解释

约束极值问题

- 岭回归的目标函数可等价地写为一个有约束的极值问题：

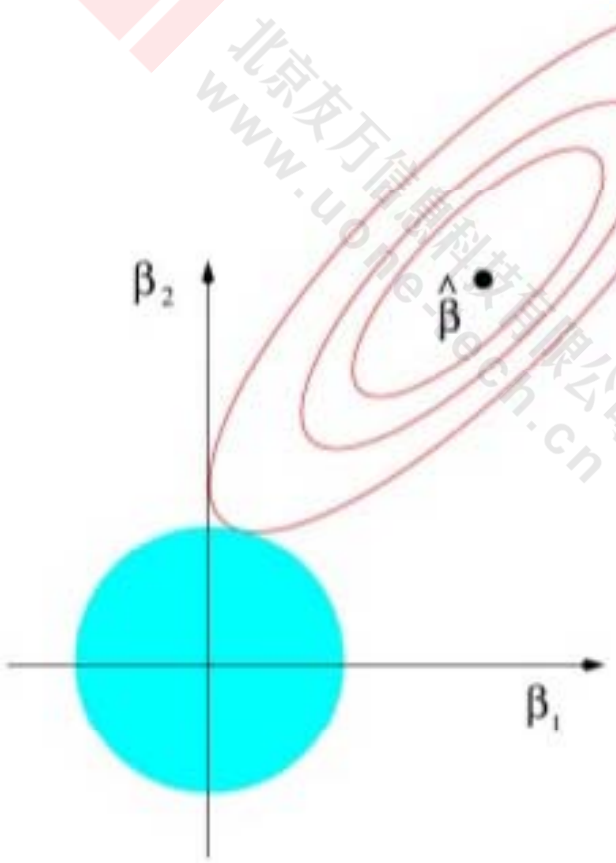
$$\min_{\boldsymbol{\beta}} \ (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})$$

$$s.t. \ \|\boldsymbol{\beta}\|_2^2 \le t$$

- 其中，$t>0$ 为某常数。对于此约束极值问题，可引入其拉格朗日乘子函数，并以 λ 作为其乘子，即可得到前述的岭回归目标函数。

岭回归图示

- 由于约束集 $\|\boldsymbol{\beta}\|_2^2 \le t$ 为p维参数空间中的圆球，故可将此约束极值问题图示如下（假设p=2）



岭回归的系数

- 由于约束集为圆球，故等高线与约束集相切的位置一般不会碰巧在坐标轴上，故岭回归通常只是将所有的回归系数都收缩，而不会让某些回归系数严格等于0。
- 在高维回归中，由于变量太多，如果所有变量的系数都非零，将使得模型的解释变得很困难
- 如何同时考察2万个回归系数？

稀疏模型

- 通常期望从2万个基因中，能够找到真正影响疾病为数不多的基因
- 一般期待真实模型是稀疏的（sparse model）
- 希望能找到一个估计量，能挑选出那些真正有影响的基因，而将其他无影响或影响微弱基因的回归系数估计为0

套索估计量

- Tibshirani (1996)提出“套索估计量”(Least Absolute Shrinkage and Selection Operator，简记 LASSO)，将岭回归的惩罚项 (也称“正则项”，regularization)中的2-范数改为1-范数

$$\hat{\boldsymbol{\beta}}_{lasso} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \ (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1$$

- $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\beta_j|$ 为参数向量 $\boldsymbol{\beta}$ 的1-范数(L_1 norm)，故称“绝对值收缩”(Absolute Shrinkage)

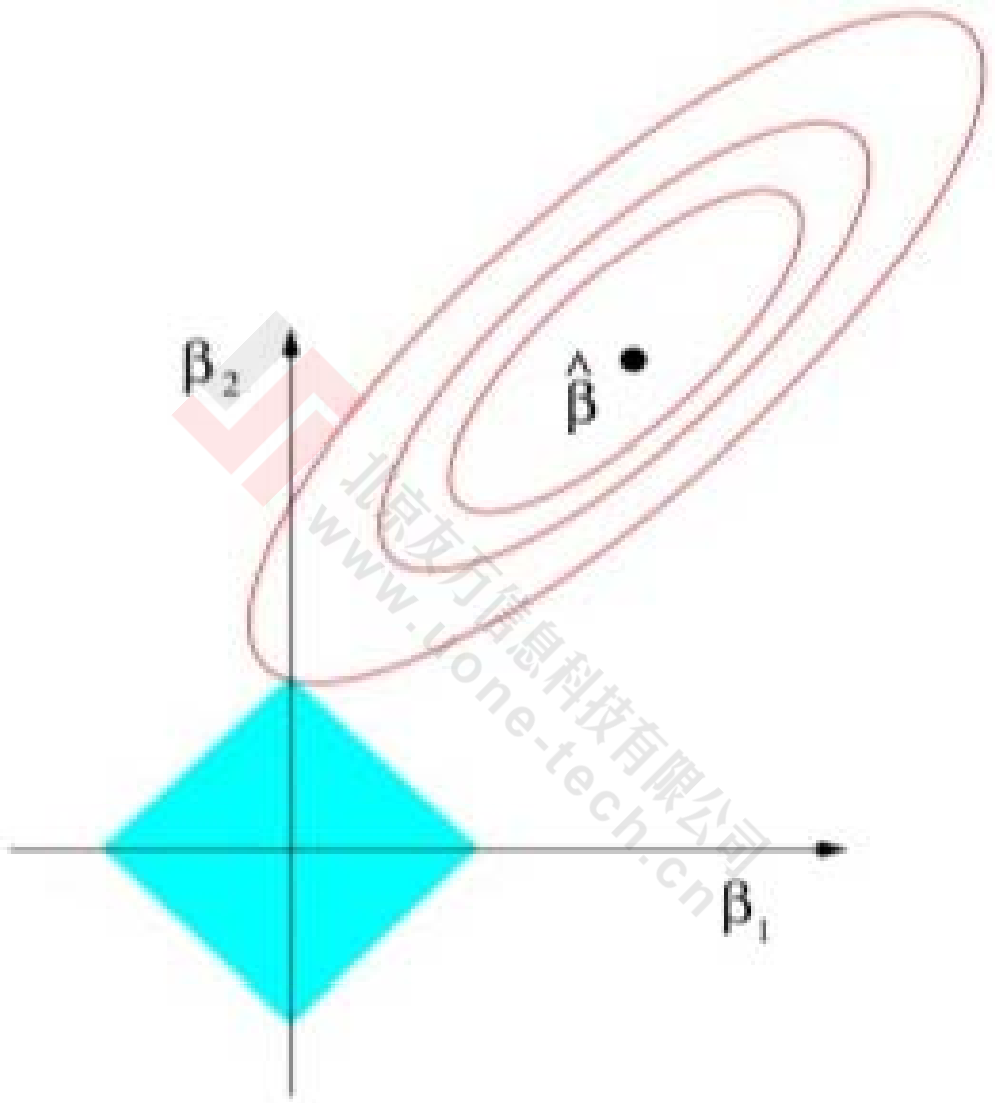
约束极值问题

- 类似于岭回归，Lasso最小化问题也可以等价地写为如下约束极值问题：

$$\min_{\boldsymbol{\beta}} \ (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})$$

$$s.t. \ \|\boldsymbol{\beta}\|_1 \le t$$

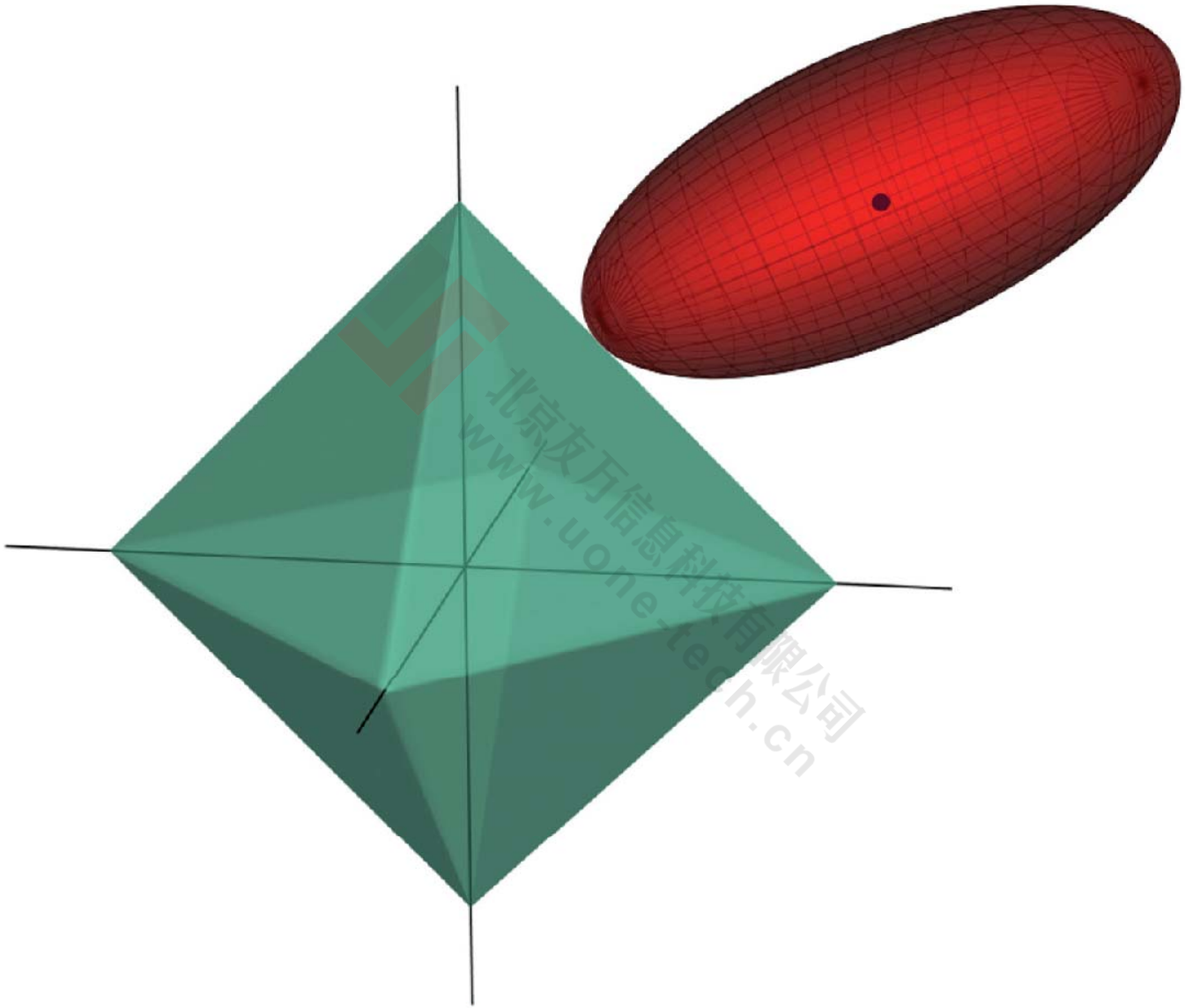
- 其中，$t>0$ 为某常数。此约束极值问题的约束集 $\|\boldsymbol{\beta}\|_1 \le t$ 不再是圆球，而是菱形或高维的菱状体



稀疏解 (Sparse Solution)

- 由于Lasso的约束集为菱形（而菱形的顶点恰好在坐标轴上），故椭圆等高线较容易与此约束集相交于坐标轴的位置，导致Lasso估计量的某些回归系数严格等于0，从而得到一个稀疏模型（sparse model）

- Lasso的这种独特性质，使得它具备了“变量筛选”（variable selection）的功能，故也称为“筛选算子”（Selection Operator）



何为 LASSO

- 综合Lasso估计量的两方面性质，故得名“最小绝对值收缩与筛选算子”(Least Absolute Shrinkage and Selection Operator，简记LASSO)

- 由于Lasso的英文原意为“套索”（想象美国西部牛仔用于套马的套索），而套索本来就有收缩之功能，故在中文译为“套索估计量”非常形象

岭回归 vs. 套索估计

- 从预测的角度看，如果真实模型（或数据生成过程）确实是稀疏的，则Lasso一般表现更优。
- 如果真实模型并不稀疏，则岭回归也可能预测效果优于Lasso。
- 从模型易于解释（interpretability）的角度，则Lasso显然是赢家，因为岭回归一般不具有变量筛选的功能。

Lasso的计算

- 由于使用了带绝对值的1-范数，Lasso的目标函数并不可微，故不存在解析解
- 由于Lasso的目标函数仍为凸函数（convex function），故存在很有效率的数值迭代算法
- 算法：Least Angle Regression (LARS)；coordinate descent algorithm (坐标下降法).

Recall: Forward Stepwise Regression Algorithm

- 这是统计学中用于选择变量 (variable selection) 的一种方法

① Start with all coefficients equal to zero.

② Find the predictor x_j most correlated with y, and add it into the model. Take residuals $r = y\text{-}yhat$.

③ Continue, at each stage adding to the model the predictor most correlated with r.

④ Until: all predictors are in the model

Least Angle Regression

- The least angle regression procedure follows the same general scheme, but doesn't add a predictor fully into the model.
- The coefficient of that predictor is increased only until that predictor is no longer the one most correlated with the residual *r*.
- LARS is state-of-the-art until 2008.

Least Angle Regression Algorithm

① Start with all coefficients equal to zero.

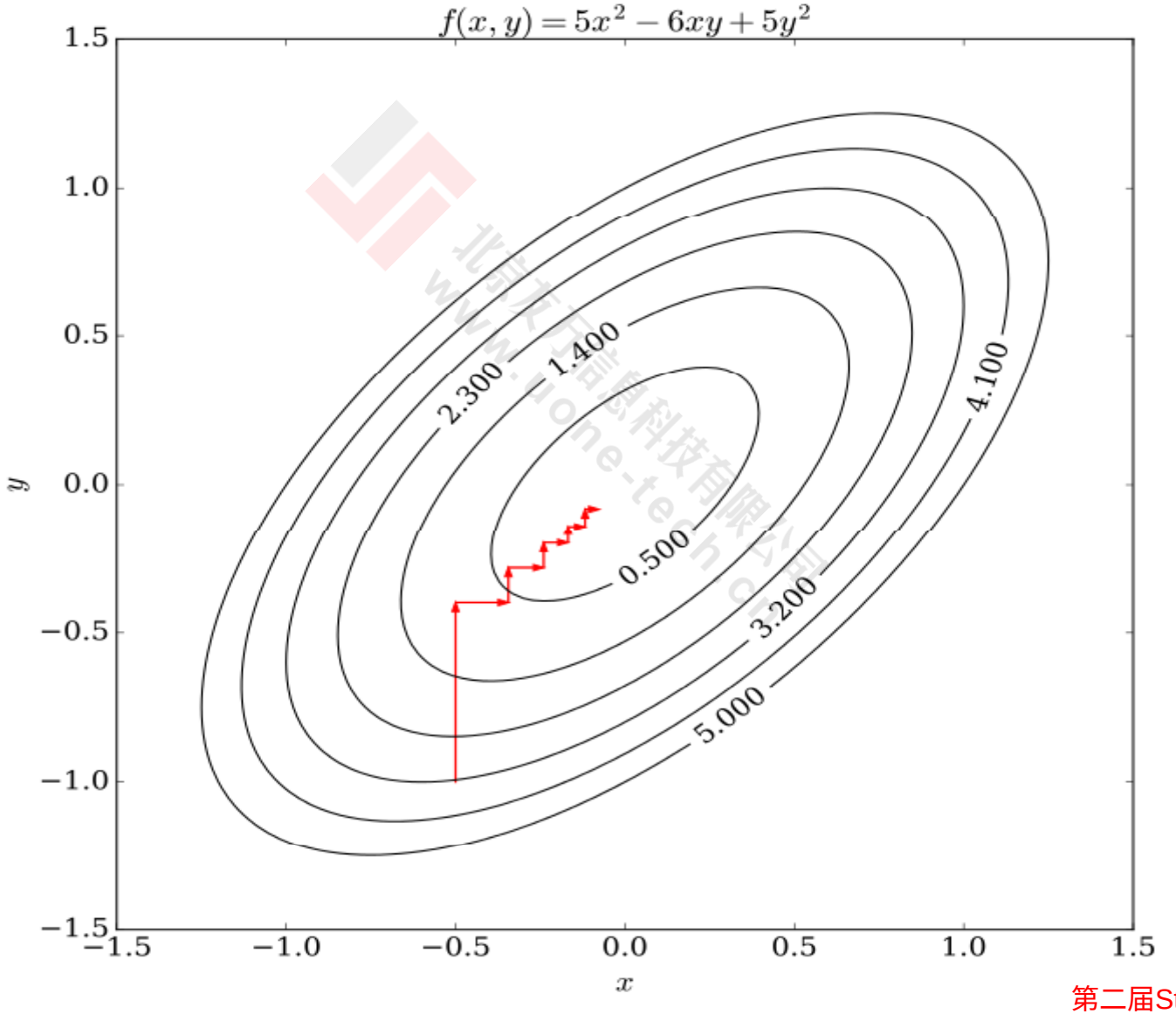
② Find the predictor x_j most correlated with y

③ Increase the coefficient b_j in the direction of the sign of its correlation with y. Take residuals $r=y-yhat$ along the way. Stop when some other predictor x_k has as much correlation with r as x_j has.

④ Increase (b_j, b_k) in their joint least squares direction, until some other predictor x_m has as much correlation with the residual r.

⑤ Continue until: all predictors are in the model

Coordinate Descent Algorithm



Pathwise coordinate descent for the lasso

- Coordinate descent: optimize one parameter (coordinate) at a time.
- How? suppose we had only one predictor. Problem is to minimize

$$\sum_i (y_i - x_i\beta)^2 + \lambda|\beta|$$

- Solution is the soft-thresholded estimate

$$\text{sign}(\hat{\beta})(|\hat{\beta}| - \lambda)_+$$

  where $\hat{\beta}$ is usual least squares estimate.
- Idea: with multiple predictors, cycle through each predictor in turn. We compute residuals $r_i = y_i - \sum_{j \neq k} x_{ij}\hat{\beta}_k$ and applying univariate soft-thresholding, pretending that our data is (x_{ij}, r_i).

Lasso的缺点（一）

- Lasso具有变量筛选的功能，但如果几个变量高度相关，则Lasso可能只选其中一个

- The lasso does not handle highly correlated variables very well; the coefficient paths tend to be erratic and can sometimes show wild behavior

- Zou and Hastie (2005)将Lasso与岭回归相结合

Elastic Net估计量

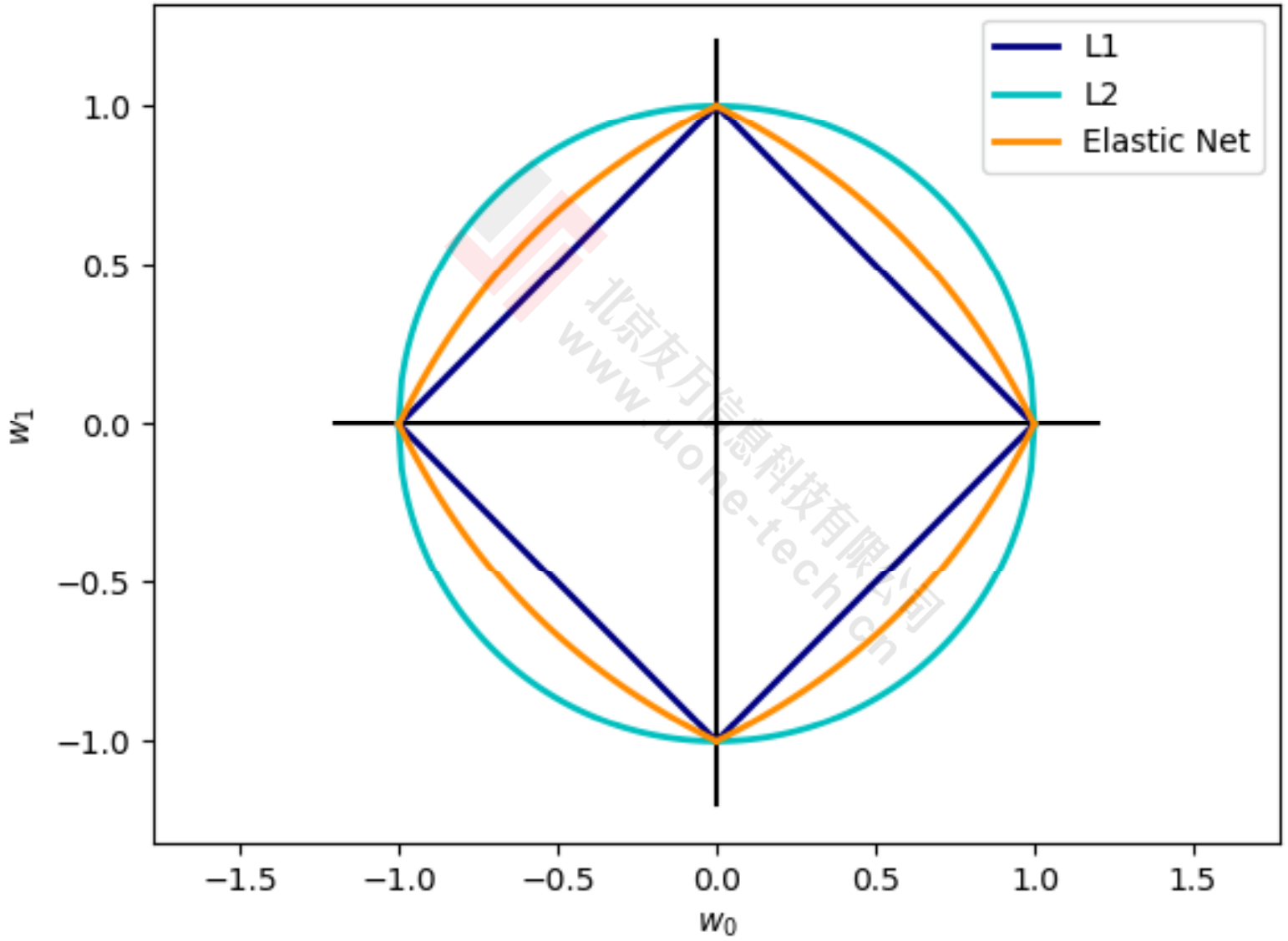
- 同时引入L_1与L_2惩罚项：

$$\hat{\boldsymbol{\beta}}_{enet} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \ (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) + \lambda_1 \|\boldsymbol{\beta}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|_2^2$$

- 令$\alpha = \lambda_1 / (\lambda_1 + \lambda_2)$（$L_1$ norm所占比重），此问题等价于

$$\min_{\boldsymbol{\beta}} \ (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})$$

$$s.t. \ \alpha \|\boldsymbol{\beta}\|_1 + (1-\alpha)\|\boldsymbol{\beta}\|_2^2 \le t$$



Elastic Net的优点

- Lasso（$\alpha=1$）与Ridge ($\alpha=0$) 都是Elastic Net的特例
- 当 *p*>*n*时，Elastic Net的预测效果可能好于Lasso
- Elastic Net encourages **grouping effects** in the presence of highly correlated predictors
- Note：Naïve elastic net suffers from double shrinkage
- Correction：$\hat{\beta}_{enet}=(1+\lambda_2)\hat{\beta}$

Lasso的缺点（二）

- Lasso不一定总能选出正确的变量：The lasso is only variable selection consistent under the rather strong "irrepresentable condition", which imposes constraints on the degree of correlation between predictors in the true model and predictors outside of the model.

- Lasso的收缩功能，使得较大的系数被压缩，导致偏差与效率损失

Oracle Properties (神谕性质)

Let us consider model estimation and variable selection in linear regression models. Suppose that $\mathbf{y} = (y_1, \ldots, y_n)^T$ is the response vector and $\mathbf{x}_j = (x_{1j}, \ldots, x_{nj})^T$, $j = 1, \ldots, p$, are the linearly independent predictors. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_p]$ be the predictor matrix. We assume that $E[y|\mathbf{x}] = \beta_1^* x_1 + \cdots + \beta_p^* x_p$. Without loss of generality, we assume that the data are centered, so the intercept is not included in the regression function. Let $\mathcal{A} = \{j : \beta_j^* \neq 0\}$ and further assume that $|\mathcal{A}| = p_0 < p$. Thus the true model depends only on a subset of the predictors. Denote by $\hat{\boldsymbol{\beta}}(\delta)$ the coefficient estimator produced by a fitting procedure δ. Using the language of Fan and Li (2001), we call δ an *oracle* procedure if $\hat{\boldsymbol{\beta}}(\delta)$ (asymptotically) has the following oracle properties:

- Identifies the right subset model, $\{j : \hat{\beta}_j \neq 0\} = \mathcal{A}$
- Has the optimal estimation rate, $\sqrt{n}(\hat{\boldsymbol{\beta}}(\delta)_{\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}^*) \rightarrow_d \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}^*)$, where $\boldsymbol{\Sigma}^*$ is the covariance matrix knowing the true subset model.

Adaptive Lasso

- Zou(2006)提出adaptive lasso，对于不同的回归系数给予不同的惩罚权重

$$\hat{\boldsymbol{\beta}}_{adaptive\ lasso} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \ (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} \hat{w}_j \left|\beta_j\right|$$

- 其中，$\hat{w}_j = \dfrac{1}{\left|\hat{\beta}_j\right|}$ (大系数所得惩罚更小)
- 而 $\hat{\beta}_j$ 为OLS估计(高维数据则用一元OLS)

Adaptive Lasso的优点

- The adaptive lasso enjoys the oracle properties.
- 在计算上与Lasso类似
- As the sample size grows, the weights for zero-coefficient predictors get inflated (to infinity), whereas the weights for nonzero coefficient predictors converge to a finite constant. Thus we can simultaneously unbiasedly (asymptotically) estimate large coefficient and small threshold estimates.

如何选择微调参数

- K-fold Cross Validation, 通常K=5, 10, 最小化MSPE

统计推断

- 目前软件仅汇报Lasso、Ridge、Elastic Net的回归系数，因为尚没有公认的标准误或*p*值
- Tibshirani(1996)提出使用自助标准误，但后来被证明是不一致的
- 机器学习的算法（algorithm）发展太快，其背后的统计推断（inference）明显滞后

如何在经济学中使用Lasso

- Lasso回归已经越来越多地出现于经济学文献中
- 但由于机器学习主要以预测为目标导向，如果简单照搬机器学习方法进行实证研究，则难免陷入误区。
- 以Lasso回归在经济学中应用为例，至少需要注意以下两方面的问题。

使用Lasso的注意事项(1)

- 第一、作为收缩估计量，Lasso是有偏的。经济学家向来不喜欢有偏估计。

- 解决方法之一为所谓“Post Lasso”估计量，即仅使用Lasso进行变量筛选，然后扔掉Lasso的回归系数，仅对筛选出来的变量进行OLS回归。

使用Lasso的注意事项(2)

- 第二、作为变量筛选算子（selection operator），Lasso并不一定就能保证避免“遗漏变量偏差”（omitted variable bias）。

- 比如，假设解释变量中包含一个我们感兴趣的处理变量（treatment variable）以及诸多控制变量（control variables）。如果直接使用Lasso估计此方程，并进行控制变量的选择，则可能忽略掉对处理变量有影响的变量（由于这些变量可能与处理变量高度相关，故在回归方程中包含处理变量的情况下，它们的作用可能被忽略），导致遗漏变量偏差。

Post Double Lasso

- 为此，Belloni, Chernozhukov and Hansen (2014, REStudy)提出了更为稳健的“Post Double Lasso”估计量

- 将被解释变量与处理变量分别对所有控制变量进行Lasso回归，然后对这两个Lasso回归（即所谓“Double Lasso”）所得的非零控制变量取并集（union）之后，再代入原方程进行OLS回归。

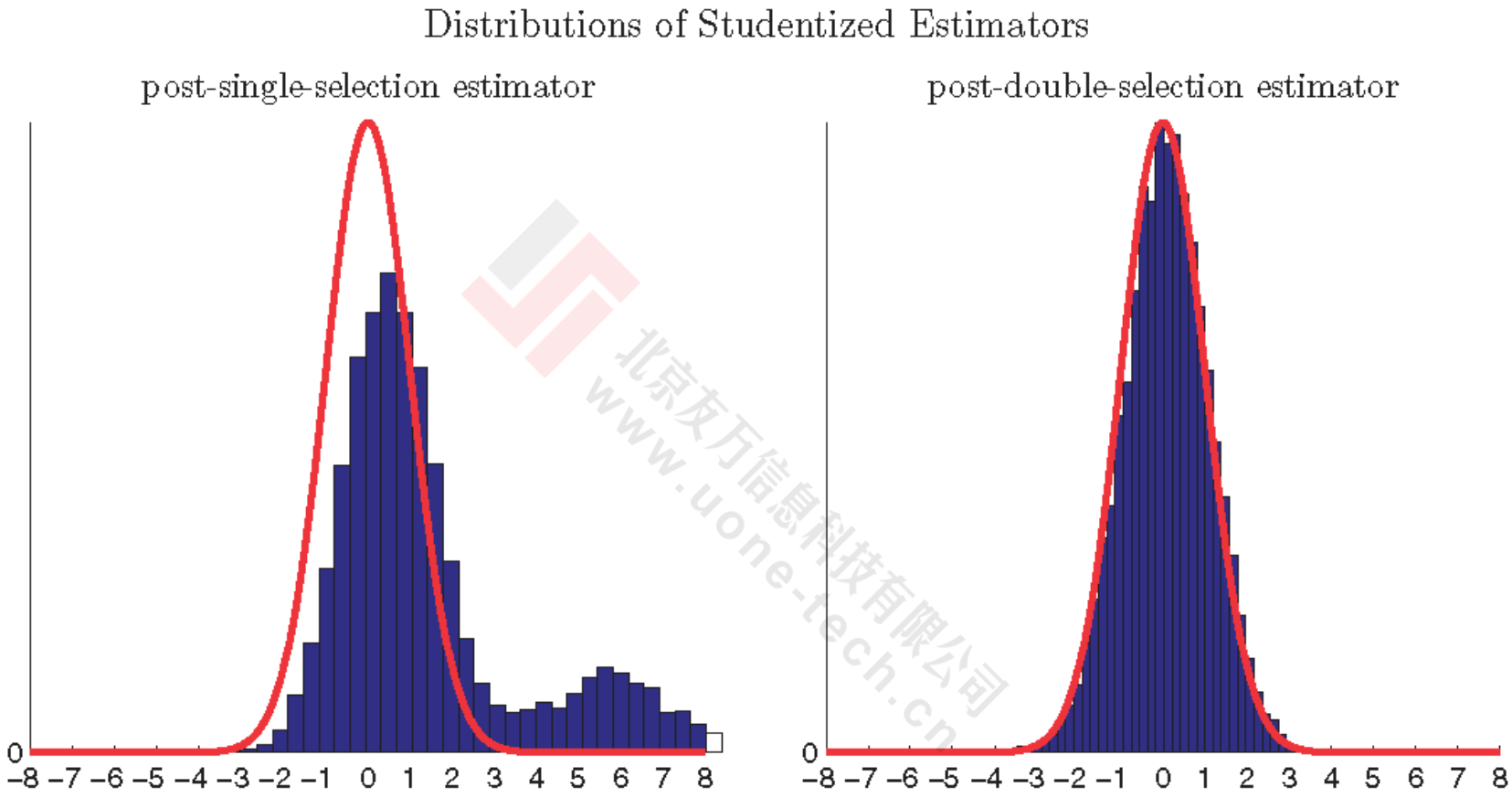


FIGURE 1

The finite-sample distributions (densities) of the standard post-single selection estimator (left panel) and of our proposed post-double selection estimator (right panel). The distributions are given for centered and studentized quantities. The results are based on 10000 replications of Design 1 described in Section 4.2, with R^2's in equation (2.6) and (2.7) set to 0.5.

IV Lasso

- 在存在很多工具变量的情况下，Belloni, Chernozhukov, Chen and Hansen (2012, Econometrica) 将Lasso应用于2SLS的第一阶段回归，以得到最优的工具变量组合。

Stata命令: elasticregress

- `ssc install elasticregress`
- `elasticregress` -- Elastic net regression
- `lassoregress` -- LASSO regression
- `ridgeregress` -- Ridge regression

Stata命令: lars

- `ssc install lars`
- 功能：Least Angle Regression, Forward Stagewise Regression, Lasso estimation
- 句型：`lars varlist,a(lasso)`
- 建议使用下文的 lassopack (功能更强)

Stata命令: lassopack

- `ssc install lassopack`
- `lasso2` implements lasso, square-root lasso, elastic net, ridge regression, adaptive lasso and post-estimation OLS.
- `cvlasso` supports K-fold cross-validation and rolling cross-validation for cross-section, panel and time-series data.
- `rlasso` implements theory-driven penalization for the lasso and square-root lasso for cross-section and panel data.

pdslasso (post double selection lasso)

- `ssc install pdslasso`
- `pdslasso` allows for estimating structural parameters in linear models with many controls
- `ivlasso` in addition allows for endogenous treatment variables and many instruments
- Needs to install `lassopack` first

Tibshirani (1996) 的例子

3. EXAMPLE—PROSTATE CANCER DATA

The prostate cancer data come from a study by Stamey *et al.* (1989) that examined the correlation between the level of prostate specific antigen and a number of clinical measures, in men who were about to receive a radical prostatectomy. The factors were log(cancer volume) (lcavol), log(prostate weight) (lweight), age, log(benign prostatic hyperplasia amount) (lbph), seminal vesicle invasion (svi), log(capsular penetration) (lcp), Gleason score (gleason) and percentage Gleason scores 4 or 5 (pgg45). We fit a linear model to log(prostate specific antigen) (lpsa) after first standardizing the predictors.

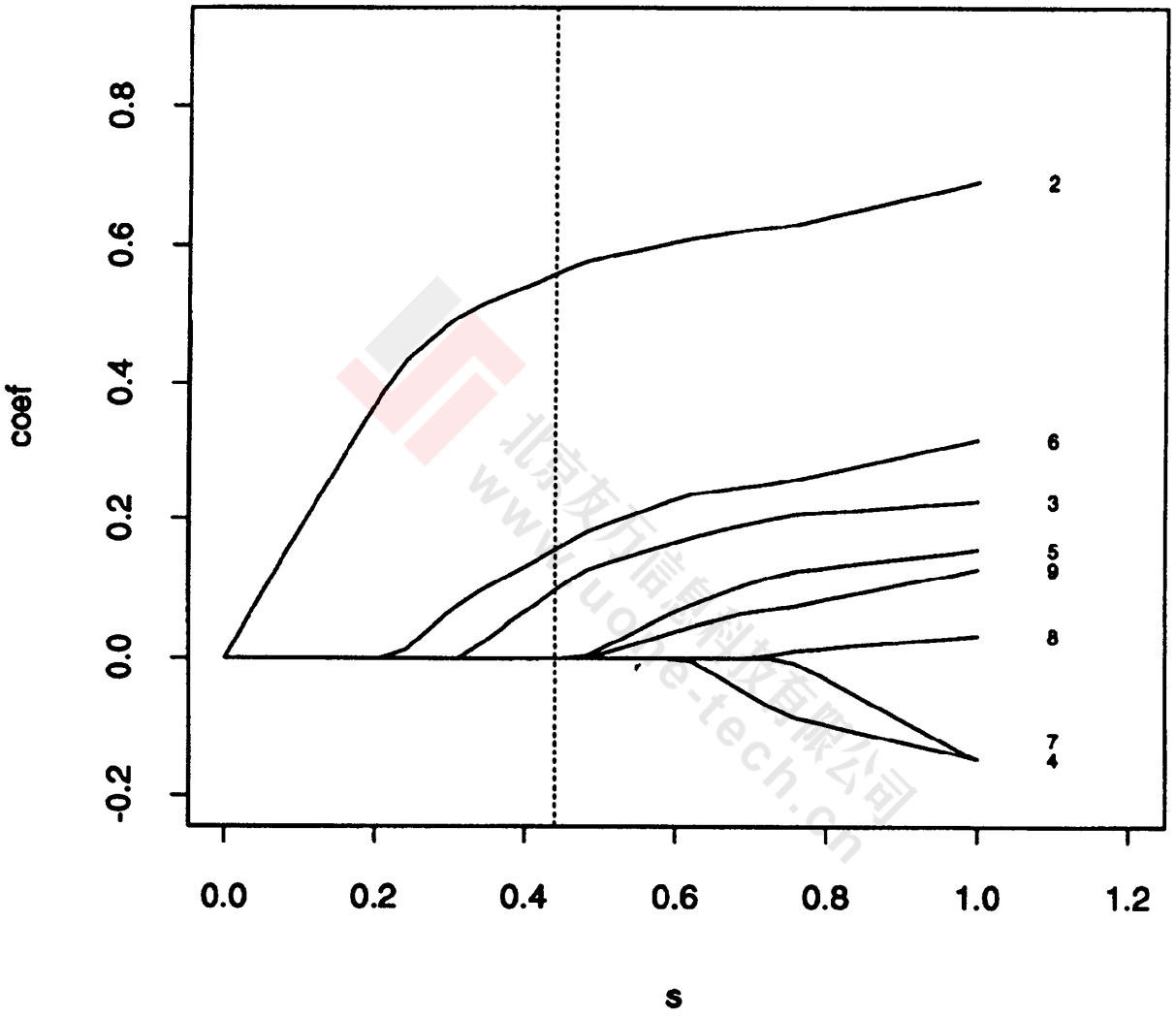


Fig. 5. Lasso shrinkage of coefficients in the prostate cancer example: each curve represents a coefficient (labelled on the right) as a function of the (scaled) lasso parameter $s = t/\Sigma|\beta_j^o|$ (the intercept is not plotted); the broken line represents the model for $\hat{s} = 0.44$, selected by generalized cross-validation

数据特征

- insheet using https://web.stanford.edu/~hastie/ElemStatLearn/datasets/prostate.data, clear tab

- sum

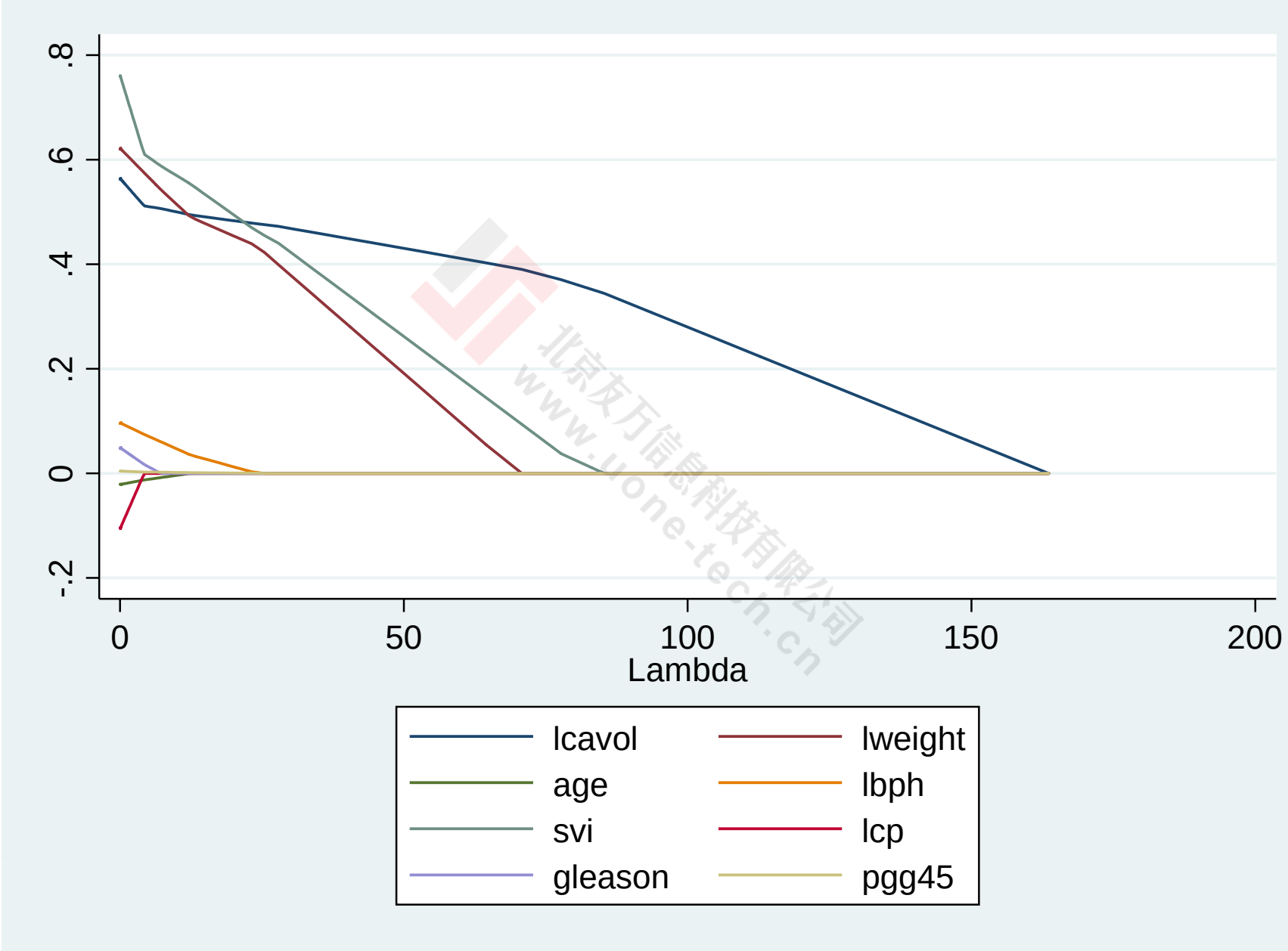
Variable	Obs	Mean	Std. Dev.	Min	Max
v1	97	49	28.14546	1	97
lcavol	97	1.35001	1.178625	-1.347074	3.821004
lweight	97	3.628943	.4284112	2.374906	4.780383
age	97	63.86598	7.445117	41	79
lbph	97	.1003556	1.450807	-1.386294	2.326302
svi	97	.2164948	.4139949	0	1
lcp	97	-.1793656	1.39825	-1.386294	2.904165
gleason	97	6.752577	.7221341	6	9
pgg45	97	24.38144	28.20403	0	100
lpsa	97	2.478387	1.154329	-.4307829	5.582932
train	0				

Solution Path of Lasso

- `lasso2 lpsa lcavol lweight age lbph svi lcp gleason pgg45, plotpath(lambda)`

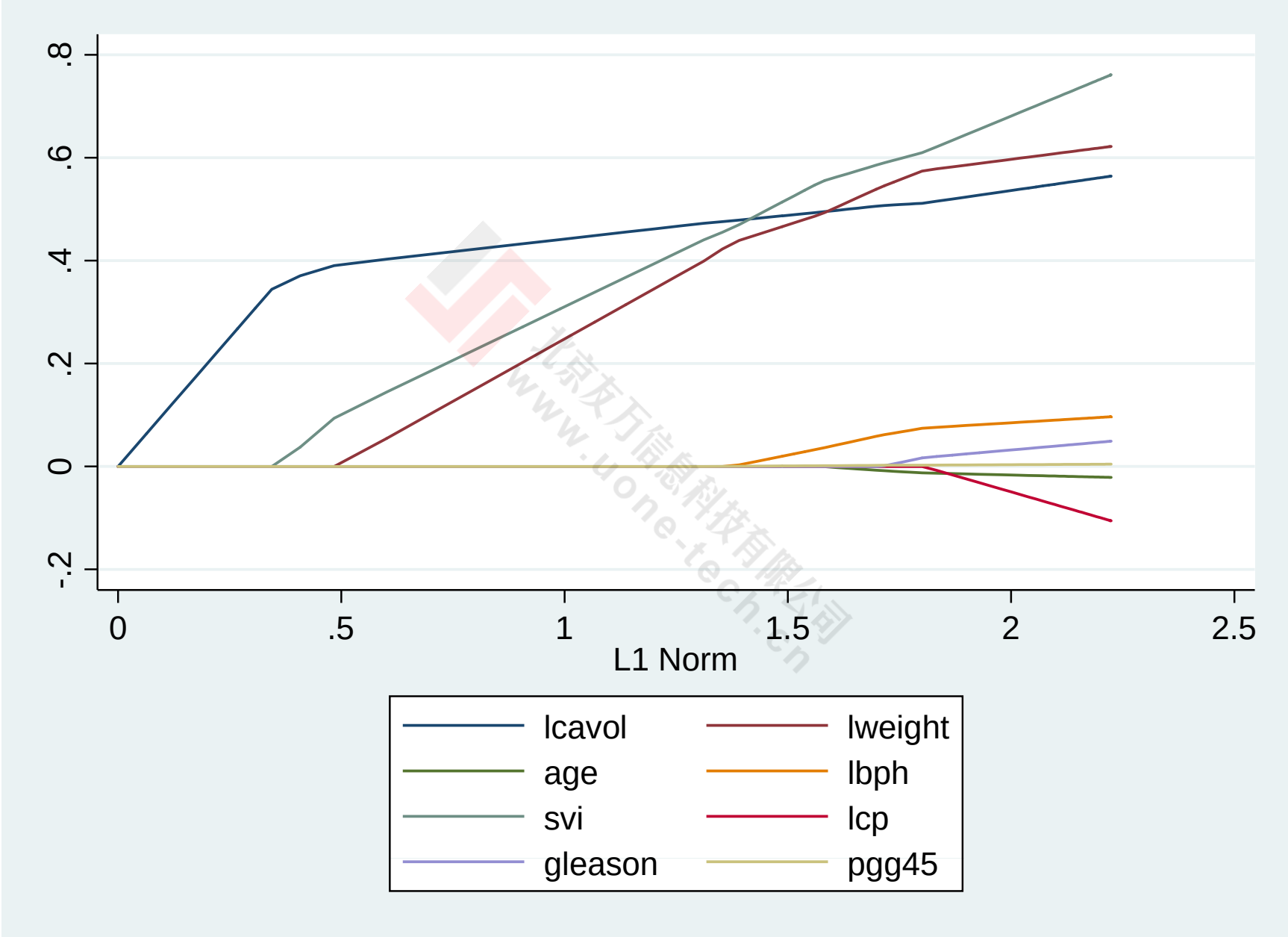
Knot	ID	Lambda	s	L1-Norm	EBIC	R-sq	Entered/removed
1	1	163.62492	1	0.00000	35.57115	0.0000	Added _cons.
2	2	149.08894	2	0.06390	34.98739	0.0043	Added lcavol.
3	9	77.73509	3	0.40800	-0.15868	0.1488	Added svi.
4	11	64.53704	4	0.60174	-1.67592	0.2001	Added lweight.
5	21	25.45474	5	1.35340	-21.40796	0.4268	Added pgg45.
6	22	23.19341	6	1.39138	-13.98342	0.4436	Added lbph.
7	29	12.09306	7	1.58269	-10.83200	0.5334	Added age.
8	35	6.92010	8	1.71689	-5.57543	0.5820	Added gleason.
9	41	3.95993	9	1.83346	1.73747	0.6130	Added lcp.

Use 'long' option for full output. Type 'lasso2, lic(ebic)' to run the model select



Solution Path of Lasso (续)

- 也可以根据L_1 norm来画图
- `lasso2 lpsa lcavol lweight age lbph svi lcp gleason pgg45, plotpath(norm)`
- 此图与Tibshirani(1996)论文中的图一致



交叉验证 (Cross Validation)

- 进行K折交叉验证 (K-fold cross validation)
- `cvlasso lpsa lcavol lweight age lbph svi lcp gleason pgg45,lopt seed(123)`
- “`lopt`”表示选择使 MSPE (Mean-Squared Prediction Error) 最小的lamda
- 默认 K=10

K-fold cross-validation with 10 folds. Elastic net with alpha=1.
Fold 1 2 3 4 5 6 7 8 9 10

	Lambda	MSPE	st. dev.	
1	163.62492	1.3162136	.13064798	
2	149.08894	1.2141972	.12282686	
3	135.84429	1.114079	.11387635	
4	123.77625	1.0312944	.10651098	
5	112.78031	.96287074	.10041915	
6	102.76122	.90634254	.09536705	
7	93.632197	.85966547	.09117793	
8	85.314171	.82129033	.0877022	
9	77.735095	.78708372	.08506697	
10	70.829323	.75291882	.08392608	
11	64.537041	.71887806	.0813633	
12	58.803749	.68700879	.07790452	
13	53.579786	.65764718	.07375266	
14	48.819905	.63330437	.07019418	
15	44.482879	.61317019	.06718914	
16	40.531143	.59652336	.06467746	
17	36.930468	.5827423	.06260056	^
18	33.649667	.5717589	.06094276	
19	30.660323	.56283992	.05958147	
20	27.936545	.55552251	.05855032	
21	25.454739	.54987747	.05788939	
22	23.19341	.5456319	.05748559	
23	21.132972	.54242135	.05725761	
24	19.255577	.53995063	.0571638	
25	17.544964	.53748677	.05733917	
26	15.986318	.53488267	.05776712	
27	14.566138	.53408884	.05830419	*
28	13.272122	.53437964	.05895557	

使用最优 λ 的Lasso最终估计结果

```
* lopt = the lambda that minimizes MSPE
^ lse = largest lambda at which the MSPE is within one standard error
Estimate lasso with lambda=14.56613752575952 (lopt).
```

Selected	Lasso	Post-est OLS
lcavol	0.4913155	0.5125769
lweight	0.4800624	0.5490809
lbph	0.0287509	0.0721513
svi	0.5367957	0.6496250
pgg45	0.0012000	0.0024450
Partialled-out*		
_cons	-0.0753696	-0.4136749

岭回归

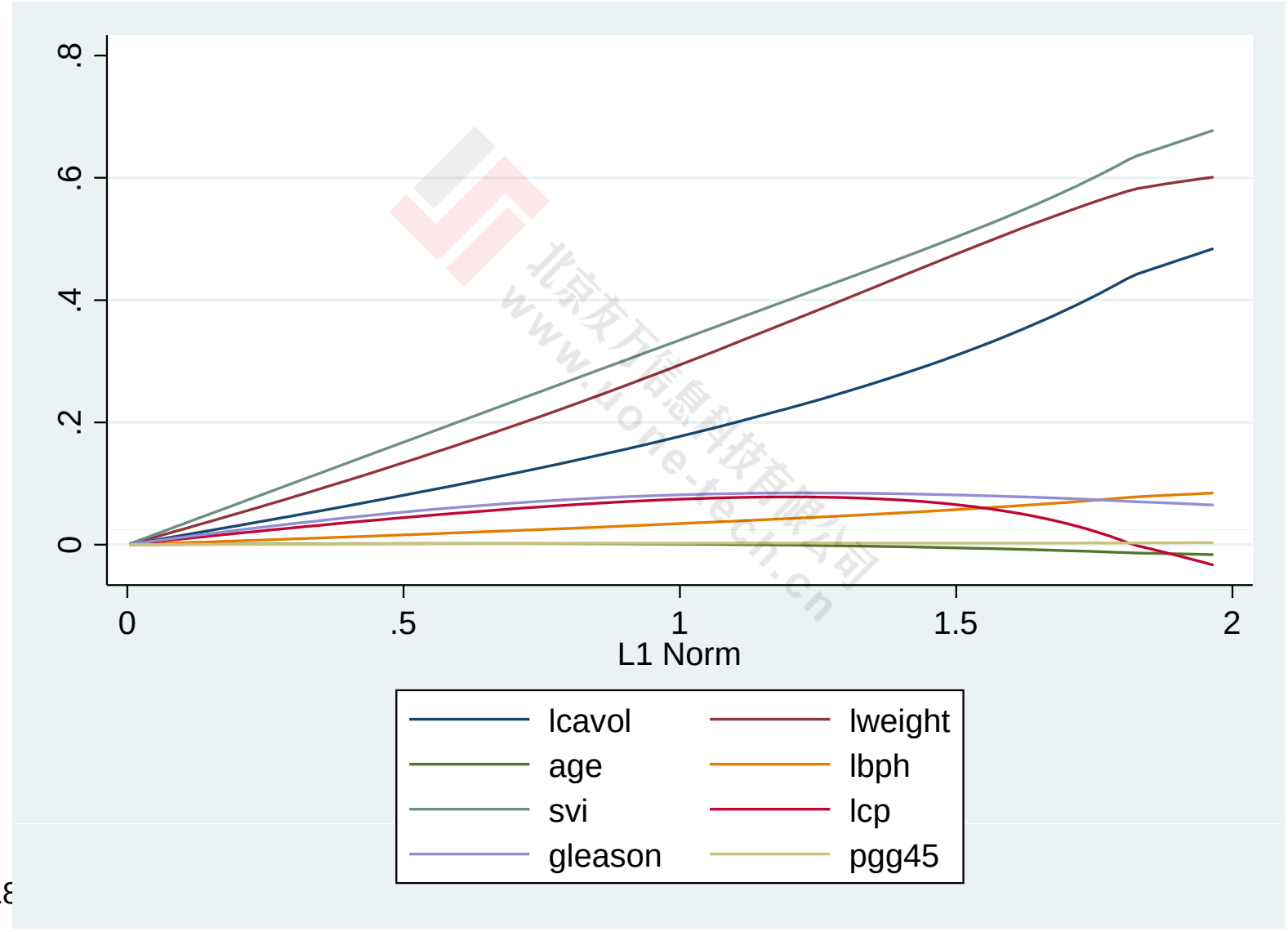
- `lasso2 lpsa lcavol lweight age lbph svi lcp gleason pgg45, plotpath(norm) alpha(0)`

- “alpha(0)”表示岭回归。默认“alpha(1)”，表示Lasso

Knot	ID	Lambda	s	L1-Norm	EBIC	R-sq	Entered/removed
1	1	1.636e+05	7	0.00550	35.30590	0.0000	Added lcavol lweight lbp lcp gleason _cons.
2	14	4.882e+04	8	0.01837	34.66794	0.0001	Added age.
3	19	3.066e+04	9	0.02916	33.93044	0.0002	Added pgg45.
4	70	266.66786	8	1.14310	-14.37223	0.2610	Removed age.
5	71	242.97782	9	1.18294	-15.15641	0.2764	Added age.

Use 'long' option for full output. Type 'lasso2, lic(ebic)' to run the model selected by E

Solution Path of Ridge



岭回归(续)：交叉验证

- `cvlasso lpsa lcavol lweight age lbph svi lcp gleason pgg45,lopt alpha(0) seed(123)`

岭回归的最终估计结果

```
Estimate ridge with lambda=12.09306327371196 (lopt).
```

Selected	Ridge	Post-est OLS
lcavol	0.5012408	0.5643413
lweight	0.6071343	0.6220198
age	-0.0173037	-0.0212482
lbph	0.0872244	0.0967125
svi	0.6949973	0.7616733
lcp	-0.0473759	-0.1060509
gleason	0.0623617	0.0492279
pgg45	0.0035150	0.0044575
Partialled-out*		
_cons	0.0290541	0.1815609

弹性网估计

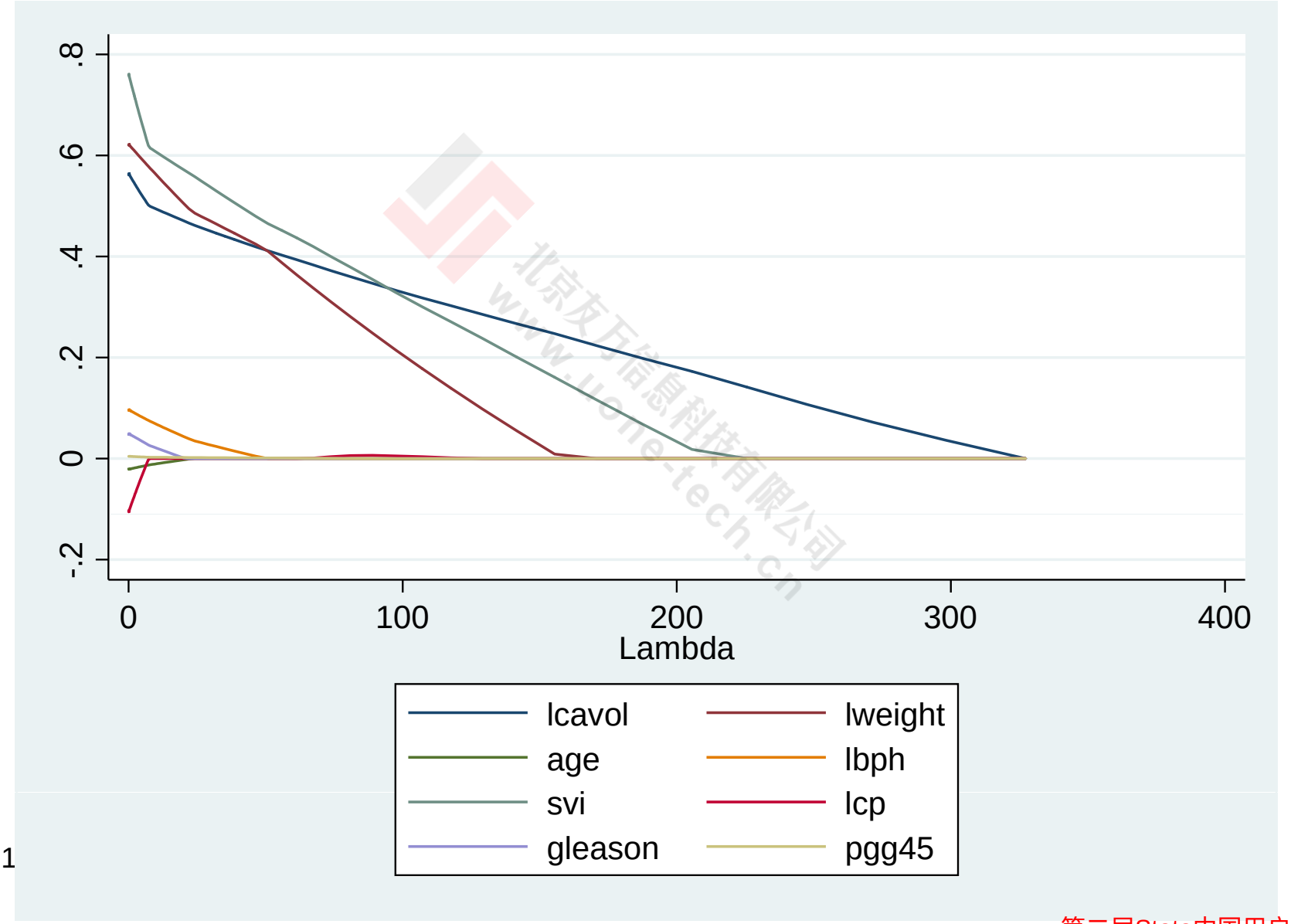
- `lasso2 lpsa lcavol lweight age lbph svi lcp gleason pgg45,alpha(0.5) plotpath(lambda)`

- “`alpha(0.5)`”表示进行弹性网估计，且$\alpha = 0.5$

Knot	ID	Lambda	s	L1-Norm	EBIC	R-sq	Entered/removed
1	1	327.24985	1	0.00000	35.57115	0.0000	Added _cons.
2	2	298.17787	2	0.03613	35.24437	0.0014	Added lcavol.
3	6	205.52244	3	0.19082	20.27745	0.0324	Added svi.
4	9	155.47019	4	0.41689	10.57230	0.0833	Added lweight.
5	12	117.60750	5	0.71407	-0.36612	0.1514	Added lcp.
6	16	81.06229	6	1.02466	-8.22341	0.2503	Added pgg45.
7	19	61.32065	5	1.19524	-20.17694	0.3206	Removed lcp.
8	22	46.38682	6	1.32918	-17.39395	0.3834	Added lbph.
9	30	22.03750	7	1.56809	-14.42549	0.5072	Added age.
10	31	20.07975	8	1.59720	-7.62706	0.5185	Added gleason.
11	42	7.21629	9	1.82024	-0.60980	0.6026	Added lcp.

Use 'long' option for full output. Type 'lasso2, lic(ebic)' to run the model selecte

Solution Path of Elastic Net



交叉验证

- cvlasso lpsa lcavol lweight age lbph svi lcp gleason pgg45,lopt alpha(0.5) seed(123)

K-fold cross-validation with 10 folds. Elastic net with alpha=.5.
Fold 1 2 3 4 5 6 7 8 9 10

	Lambda	MSPE	st. dev.	
1	163.62492	.89512771	.09696555	
2	149.08894	.84845111	.09384688	
3	135.84429	.80353514	.08934874	
4	123.77625	.76243198	.08437389	
5	112.78031	.72626652	.07962085	
6	102.76122	.69521741	.07532542	
7	93.632197	.6690077	.07152232	
8	85.314171	.64636412	.06808498	
9	77.735095	.62694021	.06518573	
10	70.829323	.61004727	.06283187	
11	64.537041	.59590251	.06086168	^
12	58.803749	.58388619	.05914503	
13	53.579786	.5740764	.05783446	
14	48.819905	.56603382	.05690078	
15	44.482879	.55946929	.05624802	
16	40.531143	.5541928	.05582761	
17	36.930468	.54889462	.05574261	
18	33.649667	.54416316	.05596494	
19	30.660323	.5401974	.05625757	
20	27.936545	.53794454	.05673325	
21	25.454739	.53714998	.05730799	
22	23.19341	.53661743	.05804729	
23	21.132972	.53607762	.05878569	
24	19.255577	.53576307	.05948455	
25	17.544964	.53522538	.06037063	
26	15.986318	.53499607	.06121615	*
27	14.566138	.53502126	.06201304	

使用最优λ的弹性网估计结果

```
Estimate elastic net with lambda=15.98631795252315 (lopt).
```

Selected	Elastic net (alpha=0.500)	Post-est OLS
lcavol	0.4801637	0.5186256
lweight	0.5279732	0.6217574
age	-0.0056509	-0.0193045
lbph	0.0534283	0.0949255
svi	0.5860216	0.6427775
gleason	0.0091456	0.0422021
pgg45	0.0022446	0.0027656
Partialled-out*		
_cons	0.0263605	0.2537330

对 α 也进行交叉验证

- `cvlasso lpsa lcavol lweight age lbph svi lcp gleason pgg45, alpha(0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1) seed(123)`
- 分别令 $\alpha = 0, 0.1, \cdots, 0.9, 1$，寻找相应的最优 λ
- 然后确定最优的 α
- 在本例中，最优 $\alpha = 1$，即Lasso

```
Cross-validation over alpha (0 .1 .2 .3 .4 .5 .6 .7 .8 .9 1).
        alpha  |  lopt*        Minimum MSPE
---------------+-------------------------------
        0.000  |  12.093063     .54348993
        0.100  |  25.454739      .5418149
        0.200  |   23.19341     .53756931
        0.300  |  19.255577     .53580978
        0.400  |  17.544964     .53525666
        0.500  |  15.986318     .53499607
        0.600  |  13.272122     .53487327
        0.700  |  12.093063     .53484357
        0.800  |  17.544964     .53491241
        0.900  |  15.986318     .53442826
        1.000  |  14.566138     .53408884  #
* lambda value that minimizes MSPE for a given alpha
# alpha value that minimizes MSPE
```

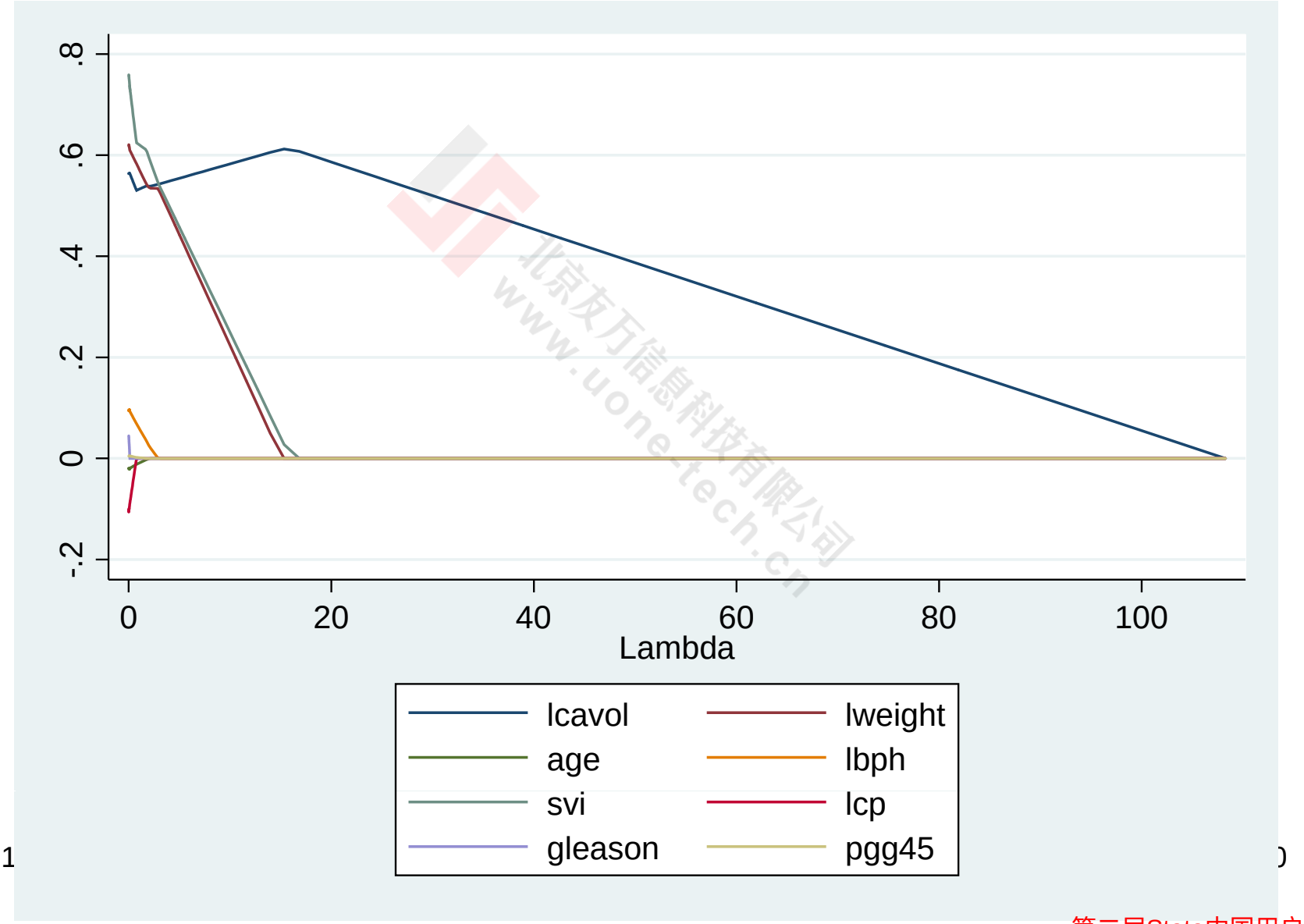
Adaptive Lasso

- `lasso2 lpsa lcavol lweight age lbph svi lcp gleason pgg45, adaptive plotpath(lambda)`

Knot	ID	Lambda	s	L1-Norm	EBIC	R-sq	Entered/removed
1	1	108.27212	1	0.00000	35.57115	0.0000	Added _cons.
2	2	98.65352	2	0.06390	34.98739	0.0043	Added lcavol.
3	22	15.34729	3	0.63951	-20.58471	0.3974	Added svi.
4	23	13.98388	4	0.73923	-15.28268	0.4101	Added lweight.
5	40	2.87580	5	1.62412	-26.57879	0.5752	Added lbph.
6	44	1.98218	6	1.69761	-19.55126	0.5940	Added age.
7	46	1.64564	7	1.73881	-12.13075	0.6021	Added pgg45.
8	55	0.71236	8	1.84669	-6.77680	0.6302	Added lcp.
9	75	0.11082	9	2.11861	-0.02777	0.6570	Added gleason.

Use 'long' option for full output. Type 'lasso2, lic(ebic)' to run the model selecte

Solution Path of Adaptive Lasso



交叉验证 (Cross Validation)

- 进行K折交叉验证 (K-fold cross validation), 默认K=10

- `cvlasso lpsa lcavol lweight age lbph svi lcp gleason pgg45,adaptive lopt seed(123)`

- “`lopt`”表示选择使 MSPE (Mean-Squared Prediction Error) 最小的lamda

K-fold cross-validation with 10 folds. Elastic net with alpha=1.
Fold 1 2 3 4 5 6 7 8 9 10

	Lambda	MSPE	st. dev.	
1	108.27212	1.3119917	.12512822	
2	98.653523	1.2262267	.11959196	
3	89.889416	1.1238329	.1105804	
4	81.903888	1.0391797	.10333121	
5	74.627772	.96922353	.09746022	
6	67.998047	.91144026	.09268124	
7	61.957288	.86373688	.08878185	
8	56.453173	.82437809	.08560331	
9	51.438029	.79192531	.08302477	
10	46.868416	.76518617	.0809516	
11	42.704755	.74317249	.0793072	
12	38.910982	.72506548	.07802752	
13	35.454238	.71018683	.07705757	
14	32.304581	.69797473	.0763493	
15	29.434731	.68796398	.07586038	
16	26.819831	.67976945	.07555346	
17	24.437231	.67307243	.07539569	
18	22.266295	.66760924	.07535846	
19	20.288218	.66326148	.07537196	
20	18.485869	.66000288	.07528454	
21	16.843635	.65683992	.07509842	
22	15.347293	.65065971	.07554494	
23	13.983882	.64035923	.0762954	
24	12.741592	.62966826	.07700607	
25	11.609665	.61444826	.0749482	
26	10.578294	.60181168	.073244	
27	9.6385474	.59141747	.07184513	^
28	8.7822855	.58280481	.07066862	
29	8.0020914	.57276518	.06848825	
30	7.2912077	.56436394	.06666549	
31	6.6434769	.55732879	.06514633	
32	6.0532887	.55162667	.06394718	
33	5.5155312	.54720102	.06305927	
34	5.0255466	.54352479	.06232443	
35	4.5790909	.5404709	.06171758	
36	4.1722971	.53793386	.06121749	
37	3.8016417	.53595281	.06085424	
38	3.4639143	.53438858	.06060621	
39	3.1561897	.53313897	.06042559	*
40	2.8758025	.53321099	.06038791	

Adaptive Lasso的最终估计结果

```
Estimate lasso with lambda=3.156189746085835 (lopt).
```

Selected	Lasso	Post-est OLS
lcavol	0.5438135	0.5258519
lweight	0.5236268	0.6617699
svi	0.5344026	0.6656665
Partialled-out*		
_cons	-0.2716736	-0.7771568