ApEc 8213: Econometric Analysis III -- Lecture #13

Machine Learning, Part 1
Hansen, Chapter 29, Sections 29.1 – 29.14
and Chapter 28, Sections 28.19 and 28.20

I. Introduction (29.1-29.2)

Machine learning methods focus on very large data sets ("**big data**"), both in terms of the number of observations and the number of variables. For example, there could be a data set with millions of observations. There could also be hundreds or thousands of variables, which is often referred to as **high dimensionality**.

Hansen describes **machine learning** as "a set of algorithmic approaches to statistical learning." It focuses on prediction. It includes:

- **Supervised learning** (prediction of Y using a large number of X variables)
- **Unsupervised learning** ("uncovering 'structure' for a large number of X variables")
- **Classification** (discrete choice analysis using a large number of predicting variables)

Hansen focuses on supervised learning.

The **vocabulary of machine learning is different from standard econometric terminology**. In particular, "training" is what econometricians call estimation, and "features, is what econometricians call regressors. **Hansen uses standard econometric terminology**.

A final point is that machine learning is "highly nonparametric", as will be seen.

II. High-Dimenional Regression and p-norms (29.3-29.4)

Consider a standard linear regression model:

$$Y = X'\beta + e$$

where X and β are $p \times 1$ vectors (Hansen usually uses K for this, but the machine learning literature uses p), with a sample size n.

We usually think of p as a relatively small number, but what happens if it is very large, even $p > n$? When $p > n$, $\hat{\beta}_{\text{ols}}$ is not uniquely defined ($\boldsymbol{X'X}$ is not full rank). Even if $p < n$, a very large p could make $\boldsymbol{X'X}$ almost singular, which result in $\hat{\beta}_{\text{ols}}$ being unstable and having high variance.

Thus, we will not use OLS but instead use various machine learning methods **when p is "very large"**.

P-Norms

Some machine learning methods (e.g ridge regression and Lasso regression) make use of "p-norms". Consider a **column vector *a* with *k* elements**. The **general definition of p-norm**, denoted by $\|a\|_p$, is:

$$\|a\|_p = \left(\sum_{j=1}^{k} |a_j|^p\right)^{1/p}$$

Note: This "p" is **not** the same p denoting number of regressors!

The **most commonly used** p-norms are $\boldsymbol{p = 1}$ **and** $\boldsymbol{p = 2}$:

$$\|a\|_1 = \sum_{j=1}^{k} |a_j|$$

$$\|a\|_2 = \left(\sum_{j=1}^{k} |a_j|^2\right)^{1/2}$$

Two other norms are **sometimes used**: $p = 0$ and $p = \infty$:

$$\|a\|_0 = \sum_{j=1}^{k} 1[a_j \neq 0]$$

$$\|a\|_\infty = \max_{1 \le j \le k} |a_j|$$

For $\|a\|_0$, any number except 0 to the power of 0 equals 1, while $0^0 = 0$, so $\boldsymbol{\|a\|_0}$ **counts the number of non-zero elements in *a*** ($\|a\|_p$ does not use "$1/p$" when $p = 0$).

For $||a||_\infty$, taking the limit of $||a||_p$ as $p \to \infty$ will lead to the largest element of a dominating all other elements of a (for example, $1.1^{40} \approx 45.3$ while $1.2^{40} = 1469.8$), so $||a||_\infty$ take the largest element to a very high power and then "returns" it using the power of one over that high power.

A p-norm applied to a vector that consists of 2 sub-vectors, a_0 and a_1, clearly has the following property:

$$(||a||_p)^p = (||a_0||_p)^p + (||a_1||_p)^p$$

Five other **properties of p-norms** are:

$$|a'b| \leq ||a||_p \times ||b||_q \text{ if } 1/p + 1/q = 1 \text{ (Hölder inequality)} \quad (29.1)$$

$$|a'b| \leq ||a||_1 \times ||b||_\infty \text{ (application of Hölder inequality)} \quad (29.2)$$

$$||a + b||_p \leq ||a||_p + ||b||_p \text{ if } p \geq 1 \text{ (Minkowski inequality)} \quad (29.3)$$

$$||a||_1 \geq ||a||_2 \geq ||a||_3 \ldots \geq ||a||_\infty \text{ if } p \geq 1 \text{ (norm monotonicity)}$$

$$||a||_1 = \Sigma_{j=1}^{k} |a_j| 1[a_j \neq 0] \leq ||a||_2 \times (||a||_0)^{1/2} \text{ (by Hölder inequality)} \quad (29.4)$$

III. Shrinkage Methods and James-Stein Shrinkage Estimator (28.19-28.20)

Shrinkage methods reduce variance in an estimator by multiplying it by a weight. Denote the **original estimator** by $\hat{\boldsymbol{\theta}}$, and **assume that it is unbiased** with a variance of $\boldsymbol{V}$, so that $\hat{\theta} \sim (\theta, \boldsymbol{V})$. Define the **shrinkage estimator** $\tilde{\boldsymbol{\theta}}$ as:

$$\tilde{\theta} = (1 - w)\,\hat{\theta}, \quad \text{where } 0 \leq w \leq 1$$

If $w > 0$, $\tilde{\theta}$ will be biased, but it may be useful because the variance is reduced. The bias and variance for $\tilde{\theta}$ are:

$$\text{bias}[\tilde{\theta}] = \text{E}[\tilde{\theta}] - \theta = -\,w\theta$$

$$\text{var}[\tilde{\theta}] = (1 - w)^2 \boldsymbol{V}$$

For **any estimator $\hat{\boldsymbol{\theta}}$ (or $\tilde{\boldsymbol{\theta}}$)**, we can define the **weighted mean squared error** (wmse) as:

$$\text{wmse}[\hat{\theta}] = \text{E}[(\hat{\theta} - \theta)'\boldsymbol{W}(\hat{\theta} - \theta)]$$

for some matrix $\boldsymbol{W}$. This is very **similar to** what we used for **density estimation** to **choose the optimal bandwidth**.

It is useful to set $\boldsymbol{W} = \boldsymbol{V}$.

Define $\lambda = \theta' \boldsymbol{V}^{-1} \theta$. We can show (could have been a good homework problem) that:

$$\text{wmse}[\tilde{\theta}] = p(1 - w)^2 + w^2 \lambda$$

where p (sometimes denoted K) = number of X variables.

Theorem 28.11. If $\hat{\theta} \sim (\theta, \boldsymbol{V})$ and $\tilde{\theta} = (1 - w)\hat{\theta}$, then:

1. $\text{wmse}[\tilde{\theta}] < \text{wmse}[\hat{\theta}]$ if $0 < w < 2K/(K + \lambda)$.

2. $\text{wmse}[\tilde{\theta}]$ is minimized by setting $w = K/(K + \lambda)$.

3. For this minimizing w, $\text{wmse}[\tilde{\theta}] = K\lambda/(K + \lambda)$.

When λ is "large", that is the coefficients θ are large compared to the variance of $\hat{\theta}$ (i.e. $\boldsymbol{V}$), then the **optimal w is "small"** (closer to 0). Conversely, **when λ is "small"**, that is the coefficients θ are small compared to the variance of $\hat{\theta}$ (i.e. $\boldsymbol{V}$), then the **optimal w is "large"** (closer to 1).

The intuition here is that when the variance of $\hat{\theta}$ is large compared to the coefficients in θ, the increase in bias from "pushing" our estimate of θ closer to 0 will be "worth it" (in terms of minimizing $\text{wmse}[\tilde{\theta}]$) because we are getting a big reduction in the variance of our estimate of θ since $\text{var}[\tilde{\theta}] = (1 - w)^2 \boldsymbol{V}$

Unfortunately, we cannot calculate the optimal w exactly because we do not know λ ($= \theta' \boldsymbol{V}^{-1} \theta$). One can show (another homework problem) that $\hat{\theta}' \boldsymbol{V}^{-1} \hat{\theta} - K$ is an unbiased estimator of λ. If we knew $\boldsymbol{V}$, we could estimate w as:

$$\hat{w} = K/(\hat{\theta}' \boldsymbol{V}^{-1} \hat{\theta})$$

It turns out (see p.915 of Hansen) that it is optimal to replace K with c, where $c = K - 2$.

Summing up, the **Stein-Rule estimator** is:

$$\tilde{\theta} = \left(1 - \frac{c}{\hat{\theta}' \boldsymbol{V}^{-1} \hat{\theta}}\right) \hat{\theta} \qquad (28.25)$$

James-Stein Shrinkage Estimator

A famous paper by James and Stein in 1961 yielded:

Theorem 28.12. Assume that $\hat{\theta} \sim \mathrm{N}(\theta, \boldsymbol{V})$, $\tilde{\theta}$ is defined as in (28.25), and $K > 2$. Then:

1. If $0 < c < 2(K-2)$, then $\mathrm{wmse}[\tilde{\theta}] < \mathrm{wmse}[\hat{\theta}]$.
2. $\mathrm{wmse}[\tilde{\theta}]$ is minimized by setting $c = K - 2$, which gives:

$$\mathrm{wmse}[\tilde{\theta}] = K - (K-2)^2 \mathrm{E}[Q_K^{-1}]$$

where $Q_K \sim \chi_K^2(\lambda)$.

This is a big shock to statisticians because $\hat{\theta}$ could be, for example, a maximum likelihood estimator, and so this estimator "dominates" ML estimation in the sense that is has a smaller wmse.

Choosing $c = K - 2$ gives the **James-Stein estimator**:

$$\tilde{\theta}_{\text{JS}} = \left(1 - \frac{K-2}{\hat{\theta}'\hat{\boldsymbol{V}}^{-1}\hat{\theta}}\right)\hat{\theta}$$

where $\widehat{\boldsymbol{V}}$ is a "good" estimator of $\boldsymbol{V}$.

IV. Ridge Regression (29.5-29.7)

Ridge regression is a **shrinkage estimator** that is **similar to**, but not the same as, the **James-Stein estimator**. Recall that when p is large, OLS has problems. The ridge regression estimator is:

$$\hat{\beta}_{\text{ridge}} = (\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I}_p)^{-1}\boldsymbol{X}'\boldsymbol{Y})$$

where $\boldsymbol{\lambda > 0}$ is the "**ridge parameter**". Unlike OLS, $\hat{\beta}_{\text{ridge}}$ is **well defined even when $\boldsymbol{p > n}$**; in particular, $\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I}_p$ has full rank (see p.944 of Hansen).

One way to derive $\widehat{\boldsymbol{\beta}}_{\text{ridge}}$ is by modifying the sum of squared residuals by **adding a "penalty", the square of**

the 2-norm. For a given λ (**not** the same λ as the shrinkage estimation λ), minimize the following with respect to β:

$$\text{SSE}_2(\beta, \lambda) = (\boldsymbol{Y} - \boldsymbol{X}\beta)'(\boldsymbol{Y} - \boldsymbol{X}\beta) + \lambda\beta'\beta$$

$$= (\|\boldsymbol{Y} - \boldsymbol{X}\beta\|_2)^2 + \lambda(\|\beta\|_2)^2$$

The **first-order condition** to minimize $\text{SSE}_2(\beta, \lambda)$ with respect to β (taking λ as fixed) is:

$$-2\boldsymbol{X}'(\boldsymbol{Y} - \boldsymbol{X}\beta) + 2\lambda\beta = 0$$

Solving for $\boldsymbol{\beta}$ gives $\widehat{\boldsymbol{\beta}}_{\textbf{ridge}}$. The **intuition** here is that **the "penalty" $\boldsymbol{\lambda\beta'\beta}$ pushes the $\boldsymbol{\beta}$ terms to be smaller**, which makes those terms "less erratic" and also "biases" the β vector toward zero.

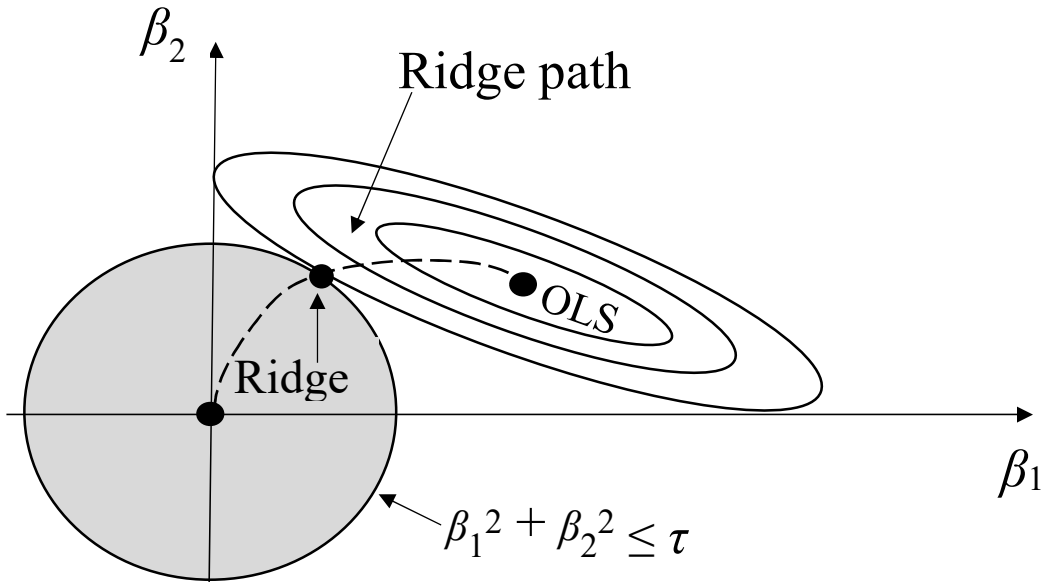
This minimization can also be achieved using constrained least squares, which constrains $\beta'\beta$ to be $\leq \tau$ for some $\tau > 0$:

$$\min_{\beta'\beta \leq \tau} \quad (\boldsymbol{Y} - \boldsymbol{X}\beta)'(\boldsymbol{Y} - \boldsymbol{X}\beta)$$

The relationship between λ and τ is the following:

$$\boldsymbol{Y}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X} + \lambda\boldsymbol{I}_p)^{-1}(\boldsymbol{X}'\boldsymbol{X} + \lambda\boldsymbol{I}_p)^{-1}\boldsymbol{X}'\boldsymbol{Y} = \tau$$

This way of expressing the ridge estimator is useful for giving a visual interpretation of what is happening, for the case of $p = 2$ (2 regressors).



The dot in the center of the concentric rings is the OLS estimate. The constraint of the ridge estimator, which is $\beta_1^2 + \beta_2^2 \leq \tau$, is the dark circle centered on the origin of β_1 and β_2. The minimized sum of squared errors that satisfied the constraint if the dot marked "Ridge". **The "Ridge Path" is the minimized sum of squared errors for different values of τ** or, equivalently, different values of λ. When $\lambda = 0$, there is no constraint and so $\hat{\beta}_{\text{ridge}} = \hat{\beta}_{\text{ols}}$. Larger values of λ move $\hat{\beta}_{\text{ridge}}$ toward the origin.

Ridge estimation can be generalized to allow different "penalties" for different groups of regressors. For example, the $\lambda\beta'\beta$ penalty can be replaced by $\lambda_1\beta_1'\beta_1 + \lambda_2\beta_2'\beta_2 + \ldots + \lambda_G\beta_G'\beta_G$, which divides β into G groups of regressors.

So how do we select λ? The "most popular" method is via **cross-validation** (**CV**). **For a given value of λ**, construct n estimates of β, each time dropping one of the n observations. The **estimate dropping observation i** is:

$$\hat{\beta}_{-i}(\lambda) = (\Sigma_{j \neq i} X_j X_j' + \lambda I_p)^{-1} \Sigma_{j \neq i} X_j Y_j'$$

Then, **use this estimate of β** to predict Y_i, and **calculate the prediction error**, denoted by $\tilde{e}_i(\lambda)$:

$$\tilde{e}_i(\lambda) = Y_i - X_i'\hat{\beta}_{-i}(\lambda)$$

Finally, **calculate the average of these squared errors**:

$$\text{CV}(\lambda) = \sum_{i=1}^{n} (\tilde{e}_i(\lambda))^2$$

Do this for different values of λ, and **choose the value of λ that minimizes CV(λ)**, that is that minimizes the out of sample prediction errors. Theorem 29.1 on page 946 of Hansen shows how to calculate $\tilde{e}_i(\lambda)$ using the residuals from the ridge regression, i.e. using $\hat{e}_i(\lambda) = Y_i - X_i'\hat{\beta}_{\text{ridge}}(\lambda)$.

An **alternative method** to select λ is to use the **Mallows criterion**, which is written as:

$$C(\lambda) = \sum_{i=1}^{n}(\hat{e}_i(\lambda))^2 + 2\hat{\sigma}^2 \text{tr}((\boldsymbol{X'X} + \lambda I_p)^{-1}(\boldsymbol{X'X}))$$

where $\hat{\sigma}^2$ is estimated using OLS. **Choose the λ that minimizes C(λ)**. If the error term of the regression model is normally distributed (which is admittedly doubtful), then the Mallows method has an optimality property.

An **important weakness** of ridge regression estimation is that the estimates can change if you rescale one or more of the regressors. **The standard practice** is to **rescale all the regressors to have a mean of 0 and a standard deviation of 1** and then apply ridge regression.

Ridge regressions can be implemented in R using "glmnet", and Stata using "lassopack".

Statistical Properties of Ridge Regression

As already mentioned, ridge regression leads to biased estimates of β. **For the standard linear regression model**, with $Y = X'\beta + e$, and $E[e|\, X] = 0$, **the bias is**:

$$\text{bias}[\hat{\beta}_{\text{ridge}}|\, \boldsymbol{X}] = -\lambda(\boldsymbol{X'X} + \lambda \boldsymbol{I}_p)^{-1}\beta$$

Under simple random sampling, the variance is:

$$\text{var}[\hat{\beta}_{\text{ridge}} | \boldsymbol{X}] = (\boldsymbol{X'X} + \lambda \boldsymbol{I}_p)^{-1}(\boldsymbol{X'DX})(\boldsymbol{X'X} + \lambda \boldsymbol{I}_p)^{-1}$$

where $\boldsymbol{D} = \text{diag}\{\sigma^2(X_1), \sigma^2(X_2), \ldots \sigma^2(X_n)\}$, where $\sigma^2(X_i) = \text{E}[e_i^2]$. For clustered data you replace $(\boldsymbol{X'DX})$ with an analogous matrix that accounts for clustered errors (Hansen does not give details).

Hansen presents Theorem 29.2 on p.947, which gives the conditions for when the mean squared error (mse) of $\hat{\beta}_{\text{ridge}}$ is less than the mse of $\hat{\beta}_{\text{ols}}$. Unfortunately, this result depends upon an unknown parameter.

In Section 29.7 on p.948, Hansen gives an empirical example that compares ridge regression to OLS.

V. Lasso (29.8 – 29.14)

We saw above that **ridge regression minimizes the sum of squared errors plus a "2-norm" penalty**. Various model selection methods (see Chapter 28 of Hansen) minimize the sum of squared errors plus a "0-norm" penalty (the penalty simply counts the number of non-zero coefficients in the model, similar to the Akaike information criterion (AIC). An "intermediate" approach is to **use a "1-norm" penalty**. This method is called

Lasso (Least Absolute Shrinkage and Selection Operator). The minimization is:

$$\text{SSE}_1(\beta, \lambda) = (\boldsymbol{Y} - \boldsymbol{X}\beta)'(\boldsymbol{Y} - \boldsymbol{X}\beta) + \lambda\Sigma_{j=1}^{p}|\beta_j|$$

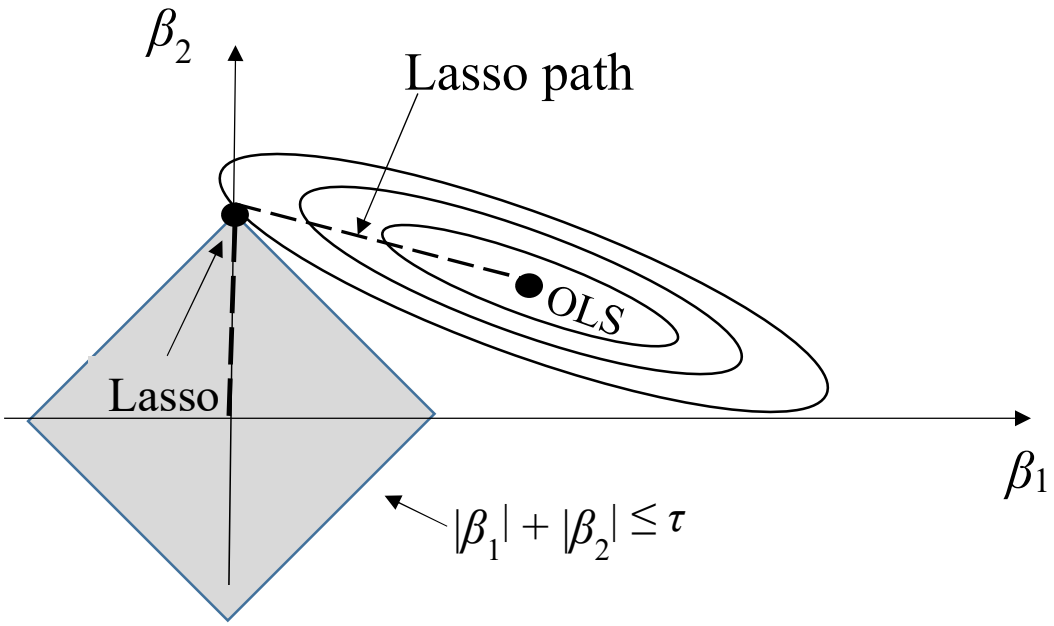
$$= (||\boldsymbol{Y} - \boldsymbol{X}\beta||_2)^2 + \lambda||\beta||_1$$

For a given value of λ, Lasso chooses β to minimize $\text{SSE}_1(\beta, \lambda)$. In general, the solution must be found using search algorithms, yet several fast algorithms have been developed. Also, when $\lambda > 0$, the Lasso estimator is well defined even when $p > n$.

As with ridge estimation, Lasso estimation **can be expressed as minimizing the sum of the squared residuals subject to some constraints**:

$$\hat{\beta}_{\text{Lasso}} = \underset{||\beta||_1 \leq \tau}{\text{argmin}}\ \text{SSE}_1(\beta)$$

This figure illustrates how Lasso works (for $p = 2$).



The constraint is indicated by the gray diamond. The Lasso estimate is the lowest SSE consistent with the constraint. A smaller value of λ (which is equivalent to a larger value of τ) relaxes this constraint.

Note that $\beta_1 = 0$, so **in effect Lasso has excluded X_1 as a regressor**. This is **common for Lasso estimation** – it excludes some regressors.

Question: Suppose that λ is reduced, which means that τ is increased. According to this figure, what will be the Lasso estimates of β_1 and β_2?

On p.950, Hansen compares the ridge and Lasso estimators and the "selection" estimator methods.

As with ridge estimation, Lasso estimation is sensitive to the scaling of the regressors, and the usual way to address this is first **rescale all variables so that they have a mean of 0 and a variance of 1**, and then apply Lasso.

Selection of λ (penalty) for Lasso

So what value of λ is optimal for Lasso estimation? As with ridge estimation, the most common method is cross-validation, although it is "**K-fold**" **cross-validation**. This is computationally much faster than the "leave-one-out" cross-validation used for ridge regression.

K-fold cross-validation is implemented as follows. First, **randomly divide the sample into *K* groups** of equal size, where ***K* is typically 5, 10 or 20**. **Estimate the Lasso model *K* times**, each time excluding one of the *K* groups. For each estimate, calculate the prediction errors *for the excluded group* by using your estimate of the model that excludes that group. Finally, square the prediction errors for all *K* groups and calculate the average of the squares of the prediction groups. **Do this for many values of λ, and choose the λ with the smallest squared prediction errors**.

In Section 29.10, Hansen describes computation of Lasso, and recommends computing the entire Lasso path (see the

figure on page 11). For R, use "glmnet", and for Stata, one can use "lasso" or "lassopack".

In Section 29.11, Hansen discusses **asymptotic theory** for Lasso, which he says is "**challenging**" and **not fully developed**. It mostly focuses on rates of convergence of estimators. This is optional material. The "sparsity" assumption, which restricts the number of parameters that can be non-zero, is a key assumption, and Section 29.12 discusses an alternative called "approximate sparsity". This is also optional.

Finally, recall that ridge regression uses a "2-norm" penalty, while Lasso uses a "1-norm" penalty. "**Elastic net**", which is described in Section 29.13, shows how to take a **weighted average of these two penalties**. This is also optional.

Post-Lasso

Lasso estimation alters OLS estimation in two ways. First, it drops some variables from the regression. Second, it "shrinks" the coefficients of the remaining variables towards zero. Since this shrinkage leads to bias, it **may be useful to use Lasso to remove "unneeded" regressors and then apply OLS** estimation to the remaining regressors, so as not to have biased estimates. This is called "**post-Lasso estimation**".

However, there are **some "problems"** with the post-Lasso estimator, since it is a "selection" estimator, such as high variance in estimated parameters and "non-standard" distributions. Hansen does not explain this in any detail.