ULB ECOLE POLYTECHNIQUE DE BRUXELLES

Computational statistics - Homework 2

High dimensional model selection

Matteo Silvestri

Professor Maarten Jansen

2024

Table des matières

1 Problem

Let's consider a high-dimensional sparse problem following the model

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \sigma \boldsymbol{Z}$$

where the design matrix $\boldsymbol{X}$ is given by the MATLAB line code

```
pX = 0.1;  X = zeros(200,1000); I = find(rand(200,1000)<pX); X(I) = (rand(size(I))–0.5)/10;
```

and the sparse vector $\boldsymbol{\beta}$ is the realization of a random variable V which has probability $1-p$ of being zero.Thus, p acts as the degree of sparsity. Given a nonzero, the distribution is taken Laplace (double exponential), which generates larger outliers than the normal errors, for clear distinction between significant and non-significant values in $\boldsymbol{\beta}$.

Discuss the experiment answering the following questions :

(a) Explain why it can be expected that the LASSO procedure over-estimates the model size.

(b) Does LASSO overestimate the size of the model in this example? (In MATLAB, you can check the size of the true model by using the command : sum(abs(beta) > 0))). If no, provide a plausible explanation.

(c) Try other design matrices, for instance by adding $\boldsymbol{X} = \boldsymbol{X} * 10$ or $\boldsymbol{X} = \dfrac{\boldsymbol{X}}{10}$ in the construction. See what happens, and explain, for instance by comparing plots μ and Y.

2 Solution

Let's first comment the main results of the MATLAB routine which can be found in the Github repository. In the figure 2.1, we have the plot of the sample Prediction Error and Cp statistic as a function of the model size k and the penalization parameter λ, where we are referring to sample PE as the value $\frac{1}{n}||\boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{X}\boldsymbol{\beta}||_2^2$. More in detail, we used 50 λ-values and calculated $\hat{\boldsymbol{\beta}}$ through the Iterative Soft-Thresholding method for each of these values. Next, the λ_{min} was calculated for these functions, and figure 2.2 shows the plot of PE and Cp as the number of iterations varies. Finally, the LARS method was applied to calculate Cp and figure 2.3 shows the comparison between the two Cp versions and it turns out that they are very comparable. Note that CpLARS stops at k=200; this is because we have n=200 observation so we can estimate at most 200 parameters (while ITST can go beyond this limit because of the regularization). It is very interesting that in this case Cp does not turn out to be an unbiased estimator of PE, although it is from a theoretical point of view. This could be related to the choice of design matrix $\boldsymbol{X}$ or may be related to the number of observations, that in this case are not so much with respect with the true model dimension (k_{true}=50).

Now let's answer to the questions.

(a) The LASSO method can overestimate the model size due to its propensity to accept many false positives because of the shrinkage, especially in scenarios involving a large number of predictors. This issue is particularly significant in sparse high-dimensional settings, where the true number of non-zero coefficients is very small. Indeed, LASSO aims to achieve sparsity by penalizing the absolute value of the regression coefficients with a regularization parameter λ and this encourages the model to set many coefficients exactly to zero. However, in practice, LASSO tends to include not just the true positives but also false positives. These are predictors that are not truly associated with the response but are selected by the model as significant.

(b) In this example, LASSO does not overestimate the size of the model that is k_{true}=50. In the following table we summarise the model size estimates by considering the abscissa of the minimum

points of the figure 2.3.

TABLE 2.1 – The optimal sizes of the model $k_{\min}$

Methods	Sample PE	Cp	CpLARS
$k_{\min}$	21	16	42

One possible explanation for this underestimation, as the professor suggested, is related to the fact that the number of observations (n=200) is not much larger than the size of the true model. This prevents the number 50 from being overestimated as 200 is already the maximum number. We merely report that this behaviour is also related to the choice of design matrix. In fact, if we take X = rand(200,1000) with appropriate rescaling(in our case is dividing by 40), then LASSO will overestimate the number of parameters giving a dimension of k=77.

(c) As we have reported, the model is sensitive to the choice of design matrix $\boldsymbol{X}$. Therefore, we want to analyse the performance of the model with respect to different design matrices. If we increase the scale of $\boldsymbol{X}$ by multiplying by 10, for instance, we amplifies the variance, which generally increases the precision of the estimated coefficients under least squares estimation. This is because the larger scale reduces the relative impact of noise on the predictor, making μ more dominant. Consequently, μ and Y may appear more aligned, reflecting a stronger signal-to-noise ratio. Conversely, reducing the scale of $\boldsymbol{X}$ dividing by 10 diminishes its variance, making the predictors weaker in relation to the noise. This can lead to a scenario where noise predominates, making it challenging to distinguish μ from the noise and leading to greater divergence between μ and Y. When testing the model with other rescaling constants, it was observed that effective results are obtained if scaling up to $\boldsymbol{X} = \boldsymbol{X} * 4$; for a rescaling-up constant ≥ 5, numerical issues arise. Whereas if scaling down even with constant 2, noise is predominant and model selection cannot be done. Figure 2.4 shows μ and Y in the standard case and the two rescaled cases with a rescaling constant of 3. To conclude our analysis, we report in figure 2.5 the results obtained using the two rescaled design matrices. As expected, the case $\boldsymbol{X} = \boldsymbol{X}/3$ does not give satisfactory results, whereas in the case $\boldsymbol{X} = \boldsymbol{X} * 3$ both Cp and CpLARS give k=55, quite close to true size.

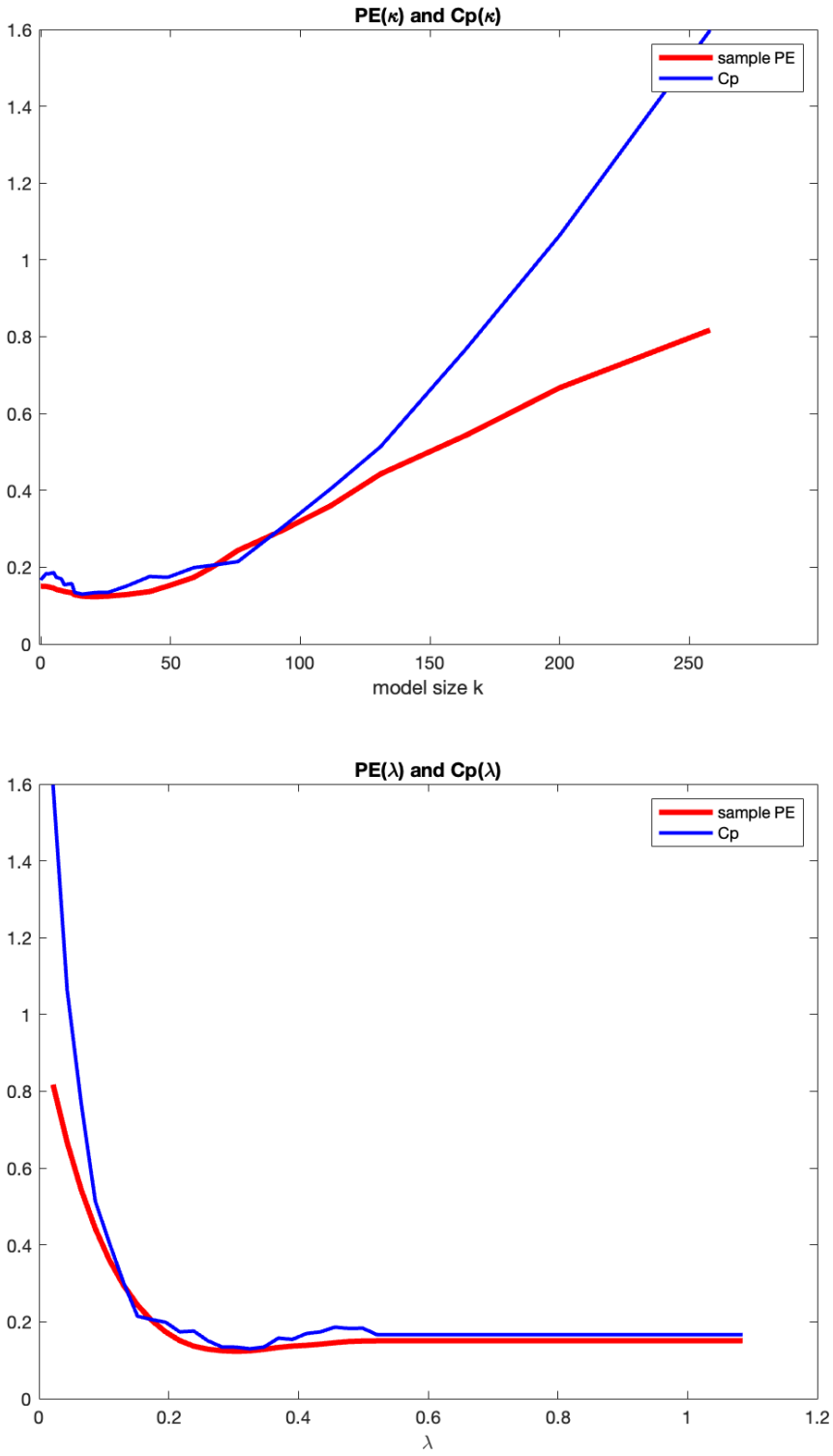

FIGURE 2.1 – Representation of PE and Cp statistic

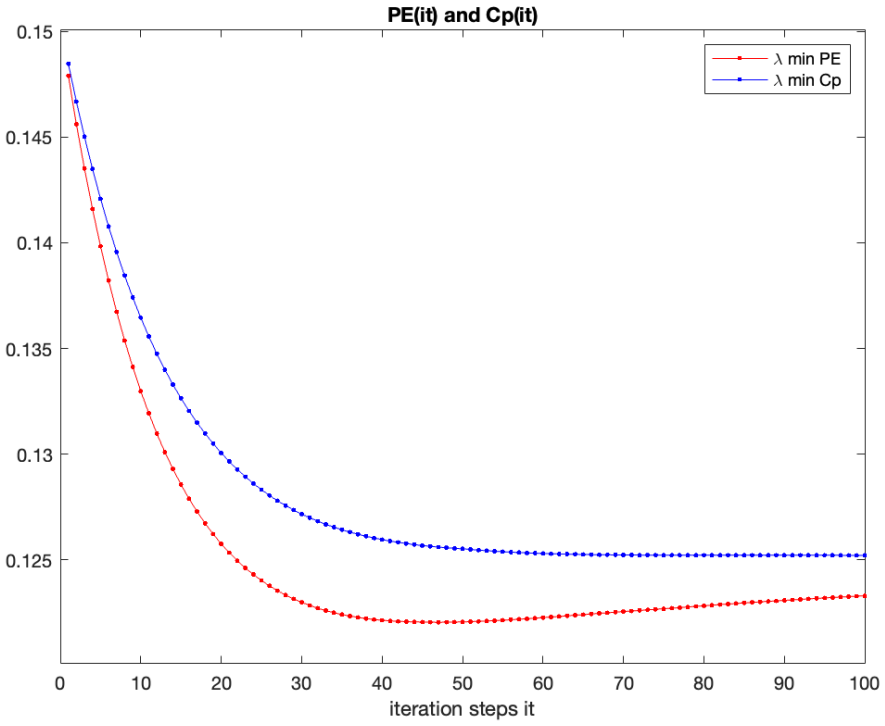

FIGURE 2.2 – Representation of PE and Cp as functions of iteration steps

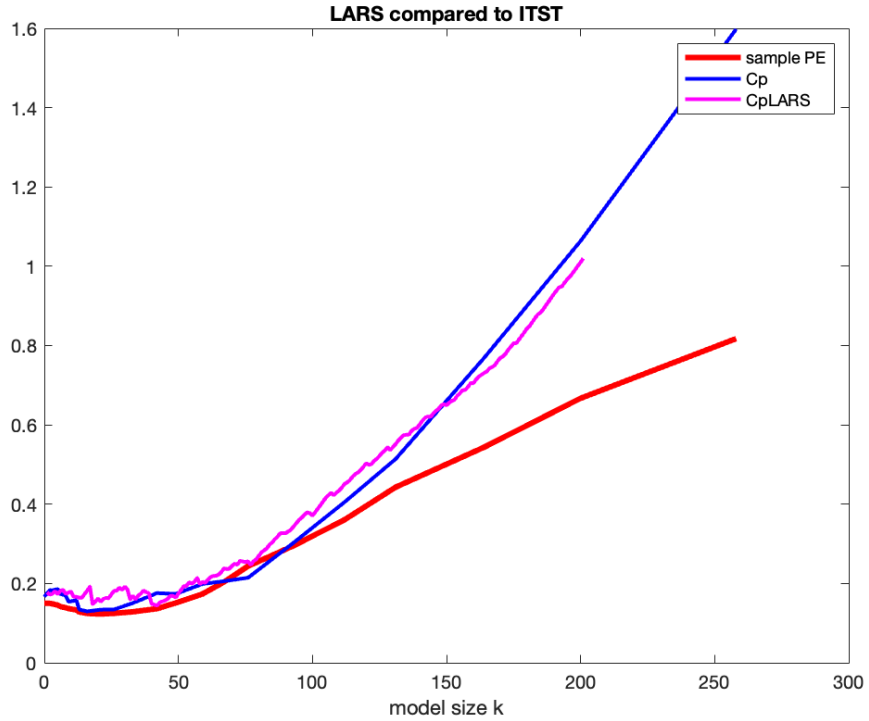

FIGURE 2.3 – Comparison between LARS and IterativeST

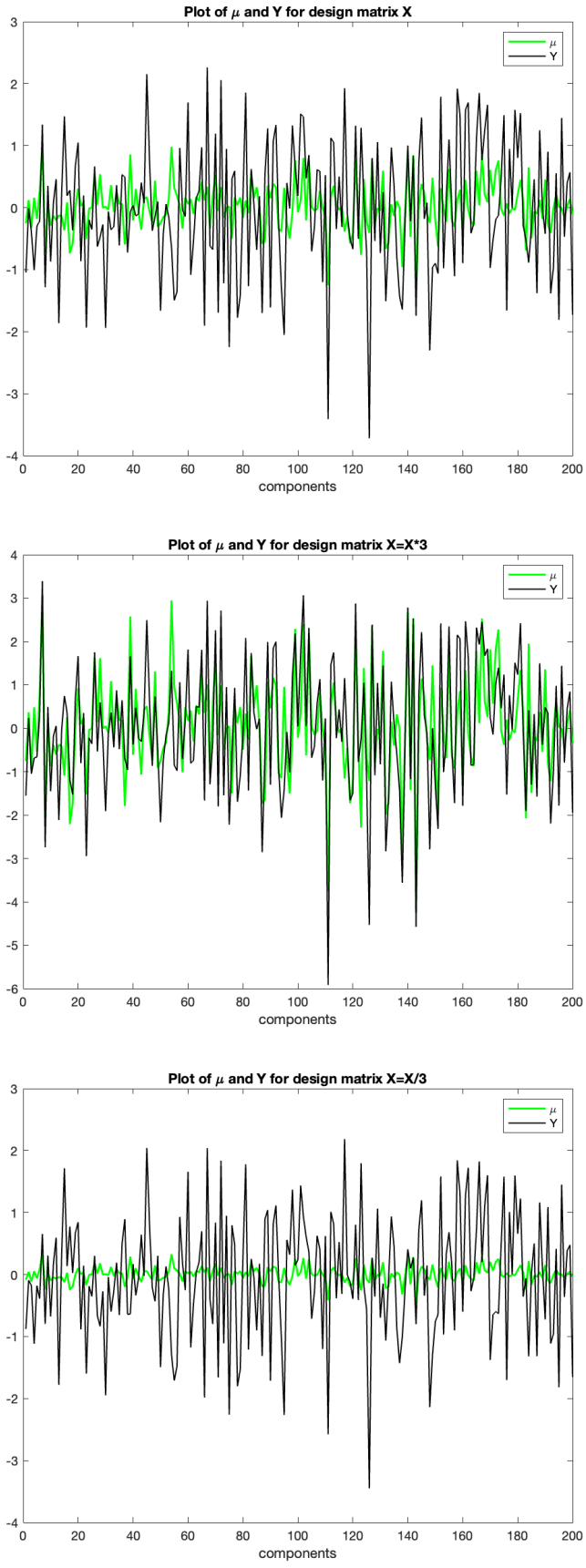

FIGURE 2.4 – Representation of μ and Y as design matrix $\boldsymbol{X}$ varies

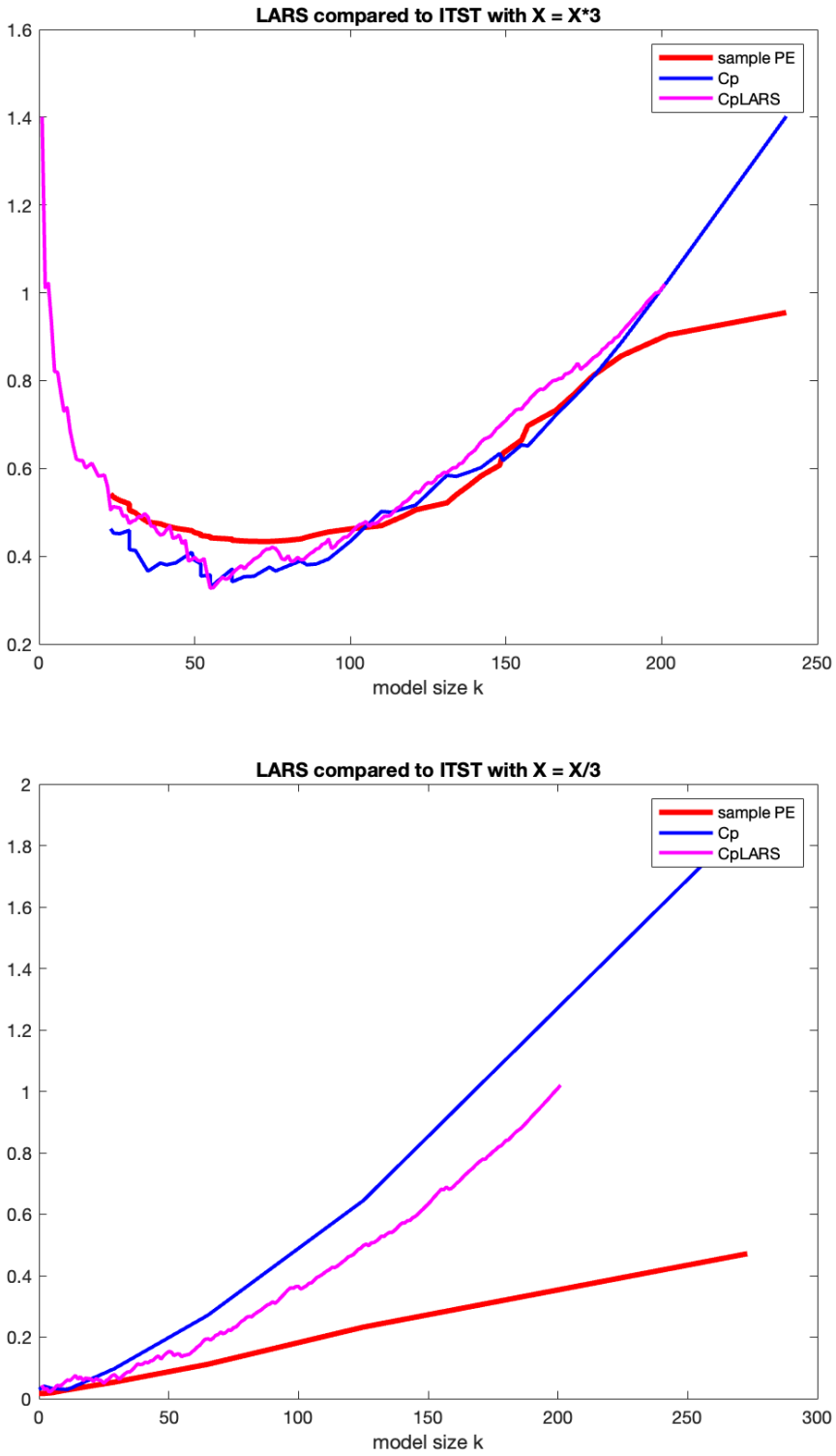

FIGURE 2.5 – Representation of PE-Cp-CpLARS as design matrix $\boldsymbol{X}$ varies