Big Data - Lecture 2

High dimensional regression with the Lasso

S. Gadat

Toulouse, Octobre 2014
Big Data - Lecture 2

*High dimensional regression with the Lasso*

S. Gadat

Toulouse, Octobre 2014
Introduction

Motivation
Trouble with large dimension
Goals
Important balance: bias-variance tradeoff

Sparse High Dimensional Regression

Sparsity
Inducing sparsity

Lasso estimation

Lasso Estimator
Solving the lasso - MM method
Statistical results

Application
In a standard linear model, we have at our disposal \((X_i, Y_i)\) supposed to be linked with

\[
Y_i = X_i^t \theta_0 + \epsilon_i, 1 \leq i \leq n.
\]

We aim to recover the unknown \(\theta_0\).

Generically, \((\epsilon_i)_{1 \leq i \leq n}\) is assumed to be i.i.d. replications of a centered and squared integrable noise

\[
\mathbb{E}[\epsilon] = 0 \quad \mathbb{E}[\epsilon^2] < \infty
\]

From a statistical point of view, we expect to find among the \(p\) variables that describe \(X\) important ones. Typical example:

- \(Y_i\) expression level of one gene on sample \(i\)
- \(X_i = (X_{i,1}, \ldots, X_{i,p})\) biological signal (DNA micro-arrays) observed on sample \(i\)
- Discover a cognitive link between DNA and the gene expression level.
Introduction - Micro-array analysis - Biological datasets

One measures micro-array datasets built from a huge amount of profile genes expression. Number of genes $p$ (of order thousands). Number of samples $n$ (of order hundred).

Diagnostic help: healthy or ill?
- Select among the genes meaningful elements?
- Find an algorithm with good prediction of the response?
From a matricial point of view, the linear model can be written as follows:

\[ Y = X\theta_0 + \epsilon, \quad Y \in \mathbb{R}^n, \ X \in \mathcal{M}_{n,p}(\mathbb{R}), \ \theta_0 \in \mathbb{R}^p \]

In this lecture, we will consider situations where \( p \) varies (typically increases) with \( n \).

\[
\begin{align*}
    y &= X \beta + \epsilon \\
    \begin{bmatrix}
        y_1 \\
        \vdots \\
        y_n
    \end{bmatrix} &= \begin{bmatrix}
        \frac{1}{\sqrt{n}} \\
        \vdots \\
        \frac{1}{\sqrt{n}}
    \end{bmatrix} \begin{bmatrix}
        \beta_1 \\
        \vdots \\
        \beta_p
    \end{bmatrix} + \begin{bmatrix}
        \epsilon_1 \\
        \vdots \\
        \epsilon_n
    \end{bmatrix}
\end{align*}
\]

\[
\begin{array}{cccc}
    n \times 1 & n \times p & p \times 1 & n \times 1
\end{array}
\]
### Introduction - Linear Model

**Standard approach:**
- \( n \gg p \)
- The M.L.E. in the Gaussian case is the Least Squares Estimator:

\[
\hat{\theta}_n := \arg \min_{\beta \in \mathbb{R}^p} \| Y - X\beta \|_2^2,
\]

given by

\[
\hat{\theta}_n = (X^t X)^{-1} X^t Y
\]

**Proposition**

\( \hat{\theta}_n \) is an unbiased estimator of \( \theta_0 \) such that

- If \( \epsilon \sim \mathcal{N}(0, \sigma^2) \):
  \[
  \frac{\| X(\hat{\theta}_n - \theta_0) \|_2}{\sigma^2} \sim \chi^2_p
  \]
- \[
  \mathbb{E} \left[ \frac{\| X(\hat{\theta}_n - \theta_0) \|_2^2}{n} \right] = \frac{\sigma^2 p}{n}
  \]
- Most of the time, \( \frac{\| X(\hat{\theta}_n - \theta_0) \|_2}{n} \) is generally negligible comparing to \( \frac{\sigma^2 p}{n} \)

**Main requirement:** \( X^t X \) must be full rank (invertible)!
I Introduction - Trouble with large dimension $p \gg n$

- $X^t X$ is an $p \times p$ matrix, but its rank is lower than $n$. If $n << p$, then

$$rk(X^t X) \leq n << p.$$ 

- Consequence: the Gram matrix $X^t X$ is not invertible and even very ill-conditionned (the most of its eigenvalues are equal to 0!)

- The linear model $\hat{\theta}_n$ completely fails.

- One standard “improvement”: use the ridge regression with an additional penalty:

$$\hat{\theta}_n^{\text{Ridge}} = \arg \min_{\beta \in \mathbb{R}^p} \|Y - X\beta\|_2^2 + \lambda\|\beta\|_2^2$$

The ridge regression is a particular case of penalized regression. The penalization is still convex w.r.t. $\beta$ and can be easily solved.

- We will attempt to describe a better suited penalized regression for high dimensional regression.

- Our goal: find a method that permits to find $\hat{\theta}_n$:
  - Select features among the $p$ variables.
  - Can be easily computed with numerical softs.
  - Possess some statistical guarantees.
Remark:

Inconsistency of the standard linear model (and even ridge regression) when \( p \gg n \).

\[
\mathbb{E} \left[ X (\hat{\theta}_n - \theta) \right] \nrightarrow 0 \quad \text{when} \quad (n, p) \xrightarrow{} +\infty \quad \text{with} \quad p \gg n.
\]

Important and nowadays questions:

- What is a good framework for high dimensional regression? A good model is required.
- How can we estimate? An efficient algorithm is necessary.
- How can we measure the performances: prediction of \( Y \)? Feature selection in \( \theta \)? What are we looking for?
- Statistical guarantees? Some mathematical theorems?
In high dimension:

- Optimize the fit to the observed data?
- Reduce the variability?

Standard question: find the best curve... In what sense?
Several regressions:
- Left: fit the best line (1-D regression)
- Middle: fit the best quadratic polynomial
- Right: fit the best 10-degree polynomial

Now I am interested in the prediction at point $x = 0.5$. What is the best?
If we are looking for the best possible fit, a high dimensional regressor will be convenient. Nevertheless, our goal is to generally to predict $y$ for new points $x$ and the matching criterion is

$$C(\hat{f}) := \mathbb{E}_{(X,Y)}[Y - \hat{f}(X)]^2.$$  

It is a quadratic loss here, and should be replaced by other criteria (in classification for example).
When the degree increases, the fit to the observed data (red curve) is always decreasing.

Over the rest of the population, the generalization error starts decreasing, and after increases.

Too simple sets of functions cannot contain the good function, and optimization over simple sets introduces a bias.

Too complex sets of functions contain the good function but are too rich and generates high variance.
The former balance is illustrated by a very simple theorem.

\[ Y = f(X) + \epsilon \quad \text{with} \quad \mathbb{E}[\epsilon] = 0. \]

**Theorem**

For any estimator \( \hat{f} \), one has

\[
C(\hat{f}) = \mathbb{E}[Y - \hat{f}(X)]^2 = \mathbb{E} \left[ Y - \mathbb{E}[\hat{f}(X)] \right]^2 + \mathbb{E} \left[ \mathbb{E}[\hat{f}(X)] - \hat{f}(X) \right]^2 + \mathbb{E} \left[ Y - f(X) \right]^2
\]

- The blue term is a bias term.
- The red term is a variance term.
- The green term is the Bayes risk and is independent on the estimator \( \hat{f} \).

**Statistical principle:**

- The empirical squared loss \( \|Y - \hat{f}(X)\|_{2,n}^2 \) mimics the bias.
- Important need to introduce something a variance control of estimation

Statistical penalty to mimic the variance.

there is an important need to control the variance of estimation.
## Schedule

1. **Introduction**
   - Motivation
   - Trouble with large dimension
   - Goals
   - Important balance: bias-variance tradeoff

2. **Sparse High Dimensional Regression**
   - Sparsity
   - Inducing sparsity

3. **Lasso estimation**
   - Lasso Estimator
   - Solving the lasso - MM method
   - Statistical results

4. **Application**
An introductory example:

- In many applications, $p > n$ but . . .
- **Important prior**: many extracted features in $X$ are irrelevant for the response $Y$.
- In an equivalent way: many coefficients in $\theta_0$ are not "almost zero" but "exactly zero".

For example, if $Y$ is the size of a tumor, it might be reasonable to suppose that it can be expressed as a linear combination of genetic information in the genome described in $X$. BUT most components of $X$ will be zero and most genes will be unimportant to predict $Y$:

- We are looking for meaningful few genes
- We are looking for the prediction of $Y$ as well.

\[
y = X \beta + \epsilon
\]

\[
\begin{array}{c}
y \\
= \\
\end{array}
\begin{array}{c}
x \\
= \\
\end{array}
\begin{array}{c}
\beta \\
+ \\
\epsilon
\end{array}
\]

\[
\begin{array}{c}
n \times 1 \\
= \\
\end{array}
\begin{array}{c}
n \times p \\
= \\
\end{array}
\begin{array}{c}
p \times 1 \\
+ \\
\text{noise}
\end{array}
\]

\[
\begin{array}{c}
p \times 1 \\
n \times 1
\end{array}
\]

S. Gadat

Big Data - Lecture 2
Dogmatic approach:

- **Sparsity**: assumption that the unknown $\theta_0$ we are looking for possesses its major coordinates null. Only $s$ of them are important:

$$s := \text{Card}\{1 \leq i \leq p|\theta_0(i) \neq 0\}.$$  

- Sparsity assumption:

$$s << n$$

- It permits to reduce the effective dimension of the problem.

- Assume that the effective support of $\theta_0$ were known, then

$$y = X \beta + \epsilon \implies y = X_S \beta_S + \epsilon$$

- If $S$ is the support of $\theta_0$, maybe $X_S^t X_S$ is full rank, and linear model can be applied.

Major issue: How could we find $S$?
**Signal processing**: in the 1990’s, how could we find for high resolution 1,2,3 dimensional signals sparse representations?

- Before going further with data: understand what they represent and try to obtain a naturally sparse representation?
- How: wavelets decomposition in signal processing.

- Sparse representation: Y. Meyer (among others)
- Efficient algorithm: S. Mallat
- Noise robustness and hard thresholding method: D. Donoho
In **statistics**: in the 2000’s, from a redundant representation, how could we find a sparse representation?

- Statistics don’t manage to improve the representation of the primary features on the data!
- Statistical estimator of the **LASSO**: R. Tibshirani, 1996.
- Efficient algorithm to solve the LASSO with the **LARS**: Efron, Johnstone, Hastie, and Tibshirani, 2002.
- Noise robustness and hard thresholding method: A. Tsybakov *et al.* (among others)

What is the LASSO method? How can we solve it? What about the statistical performances?
Ideally, we would like to find $\theta$ such that

$$\hat{\theta}_n = \arg\min_{\theta : \|\theta\|_0 \leq s} \|Y - X\theta\|_2^2,$$

meaning that the minimization is embedded in a $\ell_0$ ball.

In the previous lecture, we have seen that it is a constrained minimization problem of a convex function . . . A dual formulation is

$$\arg\min_{\theta : \|Y - X\theta\|_2 \leq \epsilon} \{\|\theta\|_0\}$$

But:

- The $\ell_0$ balls are not convex!
- The $\ell_0$ balls are not smooth!

First (illusive) idea: explore all $\ell_0$ subsets and minimize! Bullshit since:

$$C_p^s \text{ subsets and } p \text{ is large!}$$

Second idea (existing methods): run some heuristic and greedy methods to explore $\ell_0$ balls and compute an approximation of $\hat{\theta}_n$. (See next lecture)

Good idea: use a convexification of the $\|\cdot\|_0$ norm (also referred to as a convex relaxation method). How?
**Introduction**

Sparse High Dimensional Regression

**Lasso estimation**

**Application**

**Sparsity**

Inducing sparsity

\[ \ell^0 \text{ norm and convex relaxation} \]

Idea of the convex relaxation: instead of considering a variable \( z \in \{0, 1\} \), imagine that \( z \in [0, 1] \).

**Definition (Convex Envelope)**

The convex envelope \( f^* \) of a function \( f \) is the largest convex function below \( f \).

**Theorem (Envelope of \( \theta \mapsto \|\theta\|_0 \))**

- **On** \([-1, 1]^d\), the convex envelope of \( \theta \mapsto \|\theta\|_0 \) is \( \theta \mapsto \|\theta\|_1 \).
- **On** \([-R, R]^d\), the convex envelope of \( \theta \mapsto \|\theta\|_0 \) is \( \theta \mapsto \frac{\|\theta\|_1}{R} \).

**Idea:** Instead of solving the minimization problem:

\[
\forall s \in \mathbb{N} \quad \min_{\|\theta\|_0 \leq s} \|Y - X\theta\|_2^2,
\]

we are looking for

\[
\forall C > 0 \quad \min_{\|\theta\|_0^* (\theta) \leq C} \|Y - X\theta\|_2^2,
\]

**What’s new?**

- The function \( \|\cdot\|_0^* \) is convex and thus the above problem is a convex minimization problem with convex constraints.
- Since \( \|\cdot\|_0^* (\theta) \leq \|\theta\|_0 \), it is rather reasonable to obtain sparse solutions. In fact, solutions of (2) with a given \( C \) provide a lower bound of solutions of (1) with \( s \leq C \).
- If we are looking for good solutions of (1), then there must exist even better solution to (2).
$\ell^0$ norm and convex relaxation

Geometrical interpretation (in 2 D):

Left: Level sets of $\|Y - X\beta\|_2^2$ and intersection with $\ell^1$ ball. Right: Same with $\ell^2$ ball.

The left constraint problem is likely to obtain a sparse solution. Oppositely, the right constraint no!

In larger dimensions the balls are even more different:
Sparse High Dimensional Regression

Lasso estimation

Application

Sparsity

Inducing sparsity

\(\ell^1\) penalty

**Analytic point of view:** why does the \(\ell^1\) norm induce sparsity?

From the KKT conditions (see Lecture 1), it leads to a **penalized criterion**

\[
\min_{\theta \in \mathbb{R}^p : \|\theta\|_1 \leq C} \|Y - X\theta\|_2^2 \iff \min_{\theta \in \mathbb{R}^p} \|Y - X\theta\|_2^2 + \lambda \|\theta\|_1
\]

Mimics the bias

Controls the variance

In the 1d case: \(\arg \min_{\alpha \in \mathbb{R}} \frac{1}{2} |x - \alpha|^2 + \lambda |x|\): 

\[=: \varphi_\lambda(x)\]

The minimal value of \(\varphi_\lambda\) is reached at point \(x^*\) when \(0 \in \partial \varphi_\lambda(x^*)\). \(x^*\) is minimal iff 

- \(x^* \neq 0\) and \((x^* - \alpha) + \lambda \text{sgn}(x^*) = 0\).
- \(x^* = 0\) and \(d\varphi_\lambda^+(0) > 0\) and \(d\varphi_\lambda^-(0) < 0\).

**Proposition (Analytical minimization of \(\varphi_\lambda\))**

\[x^* = \text{sgn}(\alpha)[|\alpha| - \lambda]^+ = \arg \min_{x \in \mathbb{R}} \left\{ \frac{1}{2} |x - \alpha|^2 + \lambda |x| \right\}\]

For large values of \(\lambda\), the minimum of \(\varphi_\lambda\) is reached at point \(0\).
Introduction

- Motivation
- Trouble with large dimension
- Goals
- Important balance: bias-variance tradeoff

Sparse High Dimensional Regression

- Sparsity
- Inducing sparsity

Lasso estimation

- Lasso Estimator
- Solving the lasso - MM method
- Statistical results

Application
Taking all together, we introduce the *Least Absolute Shrinkage and Selection Operator* - LASSO:

\[ \forall \lambda > 0 \quad \hat{\theta}_{n}^{Lasso} = \arg \min_{\theta \in \mathbb{R}^{p}} \| Y - X\theta \|_{2}^{2} + \lambda \| \theta \|_{1} \]

The above criterion is convex w.r.t. \( \theta \).

- Efficient algorithms to solve the LASSO, even for very large \( p \).
- The minimizer may not be unique since the above criterion is not strongly convex.
- Predictions \( X \hat{\theta}_{n}^{Lasso} \) are always unique.
- \( \lambda \) is a penalty constant that must be carefully chosen.
- A large value of \( \lambda \) leads to a very sparse solution, with an important bias.
- A low value of \( \lambda \) yields overfitting with no penalization (too much important variance).
- We will see that a careful balance between \( s, n \) and \( p \) exists. These parameters as well as the variance of the noise \( \sigma^2 \) influence a "good" choice of \( \lambda \).

Alternative formulation:

\[ \hat{\theta}_{n}^{Lasso} = \arg \min_{\theta \in \mathbb{R}^{p} : \| \theta \|_{1} \leq C} \| Y - X\theta \|_{2}^{2} \]
Solving the lasso

Algorithm to solve the minimization problem $\arg \min_{\theta \in \mathbb{R}^p} \left\| Y - X\theta \right\|_2^2 + \lambda \left\| \theta \right\|_1$ is needed.

An efficient method follows the method of "Minimize Majorization" and is referred to as MM method.

- MM are useful for the minimization of a convex function/maximization of a concave one.
- Geometric illustration

Idea: Build a sequence $(\theta_k)_{k \geq 0}$ that converges to the minimum of $\varphi_{\lambda}(\theta)$.

A particular case of such a method is encountered with the E.M. algorithm useful for clustering and mixture models.

MM algorithms are powerful, especially they can convert non-differentiable problems to smooth ones.
MM algorithm

1. A function $g(\theta, \theta_k)$ is said to majorize $f$ at point $\theta_k$ if
   
   
   $$
   g(\theta_k | \theta_k) = f(\theta_k) \quad \text{and} \quad g(\theta | \theta_k) \geq f(\theta), \forall \theta \in \mathbb{R}^p.
   $$

2. Then, we define
   
   $$
   \theta_{k+1} = \arg \min_{\theta \in \mathbb{R}^p} g(\theta | \theta_k)
   $$

3. We wish to find each time a function $g(., \theta_k)$ whose minimization is easy.

4. An example with a quadratic majorizer of a non-smooth function:

5. Important remark: The MM is a descent algorithm:

   $$
   f(\theta_{k+1}) = g(\theta_{k+1} | \theta_k) + f(\theta_{k+1}) - g(\theta_{k+1} | \theta_k)
   \leq g(\theta_k | \theta_k) = f(\theta_k)
   $$

   (3)
MM algorithm for the Lasso: Coordinate descent algorithm

1. Define a sequence \((\theta_k)_{k \geq 0}\) \iff find a suitable majorization.

2. \(g : \theta \mapsto \|Y - X\theta\|^2\) is convex, whose Hessian matrix is \(X^tX\). Taylor expansion leads to

\[
\forall y \in \mathbb{R}^p \quad g(y) \leq g(x) + \langle \nabla g(x), y - x \rangle + \rho(X)\|y - x\|^2,
\]

where \(\rho(X)\) is the spectral radius of \(X\).

3. We are naturally driven to upper bound \(\varphi_\lambda\) as

\[
\varphi_\lambda(\theta) \leq \varphi_\lambda(\theta_k) + \langle \nabla g(\theta_k), \theta - \theta_k \rangle + \rho(X)\|\theta - \theta_k\|^2 + \lambda\|\theta\|_1
\]

\[
= \psi(\theta_k) + \rho(X)\left\| \theta - \left( \theta_k - \frac{\nabla g(\theta_k)}{\rho(X)} \right) \right\|_2^2 + \lambda\|\theta\|_1
\]

4. To minimize the majorization of \(\varphi_\lambda\), we then use the above proposition of soft-thresholding:

- Define

\[
\tilde{\theta}_k^j := \theta_k^j - \frac{\nabla g(\theta_k)^j}{\rho(X)}.
\]

- Compute

\[
\theta_{k+1}^j = sgn(\tilde{\theta}_k^j) \max \left[ |\theta_k^j| - \frac{2\lambda}{\rho(X)} \right]_+
\]
Importance of the results: understand difficulties from a statistical point of view.

What could we expect? In expectation or with high probability:

- **Estimation/consistency:** $\hat{\theta}_n \simeq \theta_0$.
- **Selection/Support:** $\text{Supp}(\hat{\theta}_0) \simeq \text{Supp}(\theta_0)$.
- **Prediction:** $n^{-1} \| X(\hat{\theta}_n - \theta_0) \|_2^2 \simeq s_0/n$

Statistical framework: we assume that $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ (for the sake of simplicity).

High dimensional framework:

- $s$ is the sparsity of $\theta_0$.
- $n \mapsto +\infty$ with $p = 0(e^{n\frac{1}{2} - \delta})$. It means that $p$ may be much larger than $n$.
- We are looking for a rate of convergence involving $s$, $p$ and $n$.

Important thing: choice of $\lambda$ (in terms of $s$, $p$, $n$ and $\sigma^2$).
Basic considerations (I)

We won’t provide a sharp presentation of the best known results to keep the level understandable.

Important to have in mind the extreme situation of almost orthogonal design:

\[
\frac{X^t X}{n} \simeq I_p
\]

Solving the lasso is equivalent to solving

\[
\min_w \frac{1}{2n} \|X^t y - w\|^2_2 + \lambda \|w\|_1
\]

Solutions are given by ST (Soft-Thresholding):

\[
w_j = ST_\lambda \left( \frac{1}{n} X_j^t y \right) = ST_\lambda \left( \theta_j^0 + \frac{1}{n} X_j^t \epsilon \right) =
\]
Basic considerations (II)

We would like to keep the useless coefficients to 0. It requires that

$$\lambda \geq \frac{1}{n} X^t_j \epsilon, \forall j \in J^c_0.$$  

The r.v. $\frac{1}{n} X^t_j \epsilon$ are i.i.d. with variance $\sigma^2 / n$.

The expectation of the maximum of $p - s$ Gaussian standard variables $\simeq \sqrt{2 \log(p - s)}$.

It leads to

$$\lambda = A \sigma \sqrt{\frac{\log p}{n}} , \quad \text{with} \quad A > \sqrt{2}.$$  

Precisely:

$$\mathbb{P} \left( \forall j \in J^c_0 : |X^t_j \epsilon| \leq n \lambda \right) \geq 1 - p^{1-A^2/2}.$$  

We expect that $ST_{\lambda} \mapsto Id$ to obtain a consistency result. It means that $\lambda \mapsto 0$, so that

$$\frac{\log p}{n} \mapsto 0.$$
Lasso consistency - One result

Theorem

Assume that \( \log p << n \), that all matrix \( X \) has norm 1 and \( \epsilon_i \sim \mathcal{N}(0, \sigma^2) \), then under a coherence assumption on the design matrix \( X^t X \), one has

i) With high probability, \( J(\hat{\theta}_n) \subset J_0 \).

ii) There exists \( C \) such that, with high probability,

\[
\frac{\|X(\theta_n - \theta_0)\|_2^2}{n} \leq \frac{C \sigma^2 s_0 \log p}{\kappa^2 n},
\]

where \( \kappa^2 \) is a positive constant that depends on the correlations in \( X^t X \).

One can also find results on the exact support recovery, as well as some weaker results without any coherence assumption.

N.B.: Such a coherence is measured through the almost orthogonality of the columns of \( X \). It can be traduced in terms of

\[
|\sup_{i \neq j} \langle X_i, X_j \rangle| \leq \epsilon.
\]
Short example with the R software

CRAN software: http://cran.r-project.org/web/packages/lars/

R Code:
library(lars)
data(diabetes)
attach(diabetes)
fit = lars(x,y)
plot(fit)

Lars algorithm: solves the Lasso less efficiently than the coordinate descent algorithm.

Typical output of the Lars software:
- The greater $\ell^1$ norm, the lower $\lambda$
- Sparse solution with small values of the $\|\cdot\|_1$ norm.
Removing the bias of the Lasso (I)

Signal processing example:

We have $n = 60$ noisy observations $Y(i) = f(i/n) + \epsilon_i$. $f$ is an unknown periodic function defined on $[0, 1]$, sampled at points $(i/n)$. $\epsilon_i$ are independent realizations of Gaussian r.v. We use the 50 first Fourier coefficients:

$$\varphi_0(x) = 1, \quad \varphi_{2j}(x) = \sin(2j\pi x), \quad \varphi_{2j+1}(x) = \cos(2j\pi x),$$

to approximate $f$. The OLS estimator is

$$\hat{f}^{OLS}(x) = \sum_{j=1}^{p} \hat{\beta}^{OLS}_j \varphi_j(x) \quad \text{with} \quad \hat{\beta}^{OLS} = \arg\min_{\beta} \sum_{i=1}^{n} (Y_i - \sum_{j=0}^{p} \beta_j \varphi_j(i/n))^2.$$

The OLS does not perform well on this example.
We experiment here the Lasso estimator with $\lambda = 3\sigma \sqrt{\frac{2 \log p}{n}}$ and obtain

Lasso estimator reproduces the oscillations of $f$ but these oscillations are shrunk toward 0.

When considering the initial minimization problem, the $\ell^1$ penalty select nicely the good features, but introduces also a bias (introduces a shrinkage of the parameters).

Strategy: select features with the Lasso and run an OLS estimator using the good variables.
Removing the bias of the Lasso (III)

We define
\[ \hat{f}^{\text{Gauss}} = \pi \hat{J}_0(Y) \quad \text{with} \quad \hat{J}_0 = \text{Supp}(\hat{\theta}^{\text{Lasso}}), \]

where \( \pi \hat{J}_0 \) is the \( \mathbb{L}^2 \) projection of the observations on the features selected by the Lasso.

Gauss-Lasso

\[
\beta^{\text{Adaptive Lasso}} = \arg \min_{\beta \in \mathbb{R}^p} \left\{ \|Y - X\beta\|_2^2 + \mu \sum_{j=1}^p \frac{|\beta_j|}{|\hat{\beta}_j^{\text{Gauss}}|} \right\}
\]

This minimization remains convex and the penalty term aims to mimic the \( \ell^0 \) penalty. The Adaptive Lasso is very popular and tends to select more accurately the variables than the Gauss-Lasso estimator.