# LASSO

## Study of the LASSO thanks to convex optimization tools

By: Thibault Lahire (thibault.lahire@student.isae-supaero.fr)

---

Given $x_1, ..., x_n \in \mathbb{R}^d$ data vectors and $y_1, ..., y_n \in \mathbb{R}$ observations, we are searching for regression parameters $w \in \mathbb{R}^d$ which fit data inputs to observations $y$ by minimizing their squared difference. In a high dimensional setting (when $n \ll d$) a $\ell 1$-norm penalty is often used on the regression coefficients $w$ in order to enforce sparsity of the solution (so that $w$ will only have a few non-zeros entries). Such penalization has well known statistical properties, and makes the model both more interpretable, and faster at test time.

From an optimization point of view we want to solve the following problem called LASSO (which stands for Least Absolute Shrinkage Operator and Selection Operator)

$$\text{minimize} \quad \frac{1}{2}||Xw - y||_2^2 + \lambda||w||_1 \tag{LASSO}$$

in the variable $w \in \mathbb{R}^d$, where $X = (x_1^T, ..., x_n^T) \in \mathbb{R}^{n \times d}$, $y = (y_1, ..., y_n) \in \mathbb{R}^n$ and $\lambda > 0$ is a regularization parameter.

In what follows, we derive step by step the dual of the LASSO problem and format it as a general Quadratic Problem as follows:

$$\begin{aligned} &\text{minimize} \quad v^T Q v + p^T v \\ &\text{subject to} \quad Av \preccurlyeq b \end{aligned} \tag{QP}$$

in variable $v \in \mathbb{R}^n$, where $Q \succcurlyeq 0$.

We first reformulate the problem:

$$\min_{w \in \mathbb{R}^d} \quad \frac{1}{2}||Xw - y||_2^2 + \lambda||w||_1 \quad \iff \quad \left\{ \begin{array}{c} \text{minimize} \qquad \frac{1}{2}||x||_2^2 + \lambda||w||_1 \\ w \in \mathbb{R}^d,\ x \in \mathbb{R}^n \\ \text{s.t. } x = Xw - y \end{array} \right.$$

We introduce $\nu \in \mathbb{R}^n$ the Lagrange multiplier associated to the equality constraint. We write the Lagrangian:

$$L(x, w, \nu) = \frac{1}{2}||x||_2^2 + \lambda||w||_1 + \nu^T(x - Xw + y)$$

We introduce the dual function:

$$g(\nu) = \inf_{x,w} L(x, w, \nu) = \inf_{x,w} \left\{ \frac{1}{2}x^T x + \nu^T x + \lambda||w||_1 - \nu^T X w \right\} + \nu^T y$$

$$g(\nu) = \inf_{x} \left\{ \frac{1}{2}x^T x + \nu^T x \right\} + \sup_{w} \left\{ \nu^T X w - \lambda||w||_1 \right\} + \nu^T y$$

On the one hand, we have: $h(x) = \frac{1}{2}x^T x + \nu^T x \quad \longrightarrow \nabla h(x) = x + \nu \quad \longrightarrow \nabla^2 h(x) = \mathbb{I}_{n \times n}$

$h$ is strictly convex , its minimum is obtained for $x_{\text{opt}} = -\nu$. Hence we have: $h(-\nu) = \frac{1}{2}\nu^T\nu - \nu^T\nu = -\frac{1}{2}\nu^T\nu$.
On the other hand, we set $z = X^T\nu$, we have:

$$\sup_{w} \left\{ z^T w - \lambda||w||_1 \right\} = \left\{ \begin{array}{cc} 0 & \text{if} \quad ||z||_\infty \leq \lambda \\ -\infty & \text{otherwise} \end{array} \right.$$

Hence:

$$g(\nu) = \begin{cases} \nu^T y - \frac{1}{2}\nu^T\nu & \text{if} \quad ||X^T\nu||_\infty \leq \lambda \\ -\infty & \text{otherwise} \end{cases}$$

We then write the dual problem:

$$\max_{\nu\in\mathbb{R}^n} \nu^T y - \frac{1}{2}\nu^T\nu$$
$$\text{s.t. } ||X^T\nu||_\infty \leq \lambda$$

We introduce $v = -\nu$, $Q = \frac{1}{2}\mathbb{I}_{n\times n} \succcurlyeq 0$ and $p = y$. Hence the dual problem is equivalent to:

$$\min_{v\in\mathbb{R}^n} v^T Q v + p^T v$$
$$\text{s.t. } ||X^T v||_\infty \leq \lambda$$

Moreover, $||X^T v||_\infty \leq \lambda$ means component-wise: $-\lambda\mathbb{1}_d \leq X^T v \leq \lambda\mathbb{1}_d$, i.e. $X^T v \leq \lambda\mathbb{1}_d$ and $-X^T v \leq \lambda\mathbb{1}_d$. We introduce $b = \lambda\mathbb{1}_{2d}$ and $A = [X^T, -X^T]^T$ so that $||X^T v||_\infty \leq \lambda$ is equivalent to $Av \preccurlyeq b$. We then have the problem:

$$\min_{v\in\mathbb{R}^n} v^T Q v + p^T v$$
$$\text{s.t. } Av \preccurlyeq b$$

In the python code along with this PDF, the barrier method solving the QP problem is implemented. The function centering_step implements the Newton method to solve the centering step and the function barr_method implements the barrier method to solve QP using the precedent function. There are two simulations:

- The first one, implemented in the function first_tests, is here to test that the method works properly.
- The second one, implemented in the function further_tests, tests the method on randomly generated matrices $X$ and observations $y$ with $\lambda = 10$.

The variable toy_problem executes first_tests when set to one, further_tests otherwise.

Set toy_problem = 1.

We study an example where $X = [1, 2]^T$ and $y = [1, 2]^T$, $d = 1$, $n = 2$. If we were solving $\min_\beta ||X\beta - y||_2^2$, $\beta = 1$ is the obvious solution. But we are solving the LASSO problem through a constrained QP problem where $A = [[1, 2], [-1, -2]]$ and $b = [\lambda, \lambda]^T = [10, 10]^T$. The feasible set of the QP can be seen on Fig. 1.

We observe that the solution $v_{\text{opt}}$ of the QP problem is inside the feasible set and that $w_{\text{opt}}$ is very close to zero. Since $\lambda$ is large for the values of this problem, the constraint of the QP problem is useless. Moreover, with $w_{\text{opt}}$ very close to zero, we see that the algorithm tries to minimize as best as possible the "huge" regularization term $\lambda||w||_1$ in the expression $\frac{1}{2}||Xw - y||_2^2 + \lambda||w||_1$.

Now, change lines 40, 41, and 42 to have $X = [10, 20]^T$ and $y = [10, 20]^T$ (or see Fig. 2).

We observe that the solution $v_{\text{opt}}$ of the QP problem is very close to the frontier of the feasible set, and $w_{\text{opt}}$ is close to one. For the values of this problem, $\lambda$ is not so large and the constraints of the QP problem really matters. With $w_{\text{opt}}$ close to one, we see that the algorithm tries to minimize as best as possible the term $\frac{1}{2}||Xw - y||_2^2$ in the expression $\frac{1}{2}||Xw - y||_2^2 + \lambda||w||_1$.

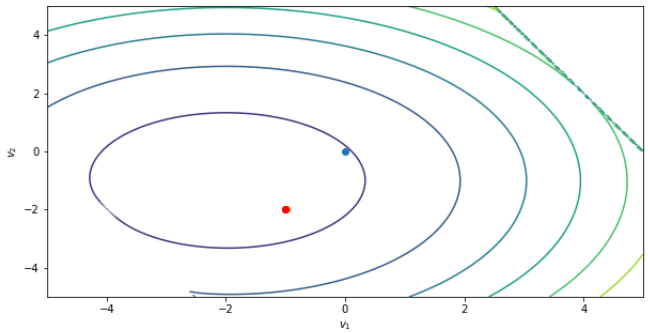


Figure 1: The red point corresponds to the QP optimal point, whereas the blue point is the starting point. We took $X = [1, 2]^T$ and $y = [1, 2]^T$

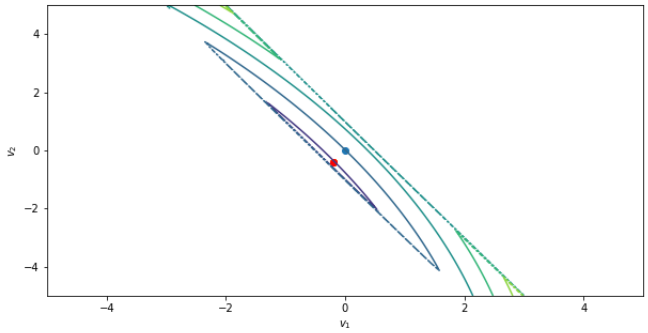


Figure 2: Same caption as Fig. 1. The feasible set is more visible here. We took $X = [10, 20]^T$ and $y = [10, 20]^T$

Minimizing $\lambda||w||_1$ is not a priority.

Now, set toy_problem to a value different than one.

We always choose v0 = zeros because we know that this point is feasible. We initialize t = 10 and we want a precision of $10^{-6}$.

The problem the algorithm solves has for cost function $t(v^T Qv + p^T v) + \phi(v)$ with $\phi(v) = -\sum_i \log((b - Av)_i)$ . The formula for the gradient is: $\nabla = t(2Qv + p) + \sum_i \frac{1}{b_i - A_i v} A_i$ and Hessian $\nabla^2 = 2tQ + \sum_i \frac{1}{(b_i - A_i v)^2} A_i A_i^T$.

We test the algorithm for different values of $\mu$ (barrier method parameter). A low value of $\mu$ leads to few Newton inner steps but many outer steps. (" $t$ goes slowly to infinity"). A high value of $\mu$ leads to many Newton inner steps but just a few outer steps.

In theory, it is not recommended to start the algorithm with a very large $t$ (or a very large $\mu$), because this means a bad conditioned problem. However, for the values tested in this problem, there is no real difference for values of $\mu$ larger than 15 (see Fig. 3). It is a well-known result that $\mu$ is a hyper-parameter easy to tune: choose $\mu = 20$, and the algorithm will work well.

We finish this report by expliciting the way we retrieve $w^*$ solving the LASSO problem from $v^*$ optimizing the QP dual.

We are solving numerically the QP dual, so let's look at the Lagrange multipliers of the QP dual. We need to compute the dual of the QP dual.

$$\begin{cases} \min\limits_{v\in\mathbb{R}^n} \frac{1}{2}v^Tv + p^Tv \\ \text{s.t. } Av \preccurlyeq b \end{cases} \iff \begin{cases} \min\limits_{v\in\mathbb{R}^n} \frac{1}{2}v^Tv + y^Tv \\ \text{s.t. } X^Tv \preccurlyeq \lambda \mathbb{1}_d \quad \text{and} \quad -X^Tv \preccurlyeq \lambda \mathbb{1}_d \end{cases}$$

Hence, the Lagrangian has the following form:

$$\begin{aligned} L(v, w_1, w_2) &= \frac{1}{2}v^Tv + y^Tv + w_1^T(X^Tv - \lambda\mathbb{1}_d) + w_2^T(-X^Tv - \lambda\mathbb{1}_d) \\ &= \frac{1}{2}v^Tv + (y^T + w_1^TX^T - w_2^TX^T)v - \lambda\mathbb{1}_d^T(w_1 + w_2) \end{aligned}$$

We introduce $h(v) = \frac{1}{2}v^Tv + (y + Xw_1 + Xw_2)^Tv$, note that $h$ is strictly convex.

$$0 = \nabla h(v_{\text{opt}}) = v + y + Xw_1 + Xw_2 \quad \iff \quad v_{\text{opt}} = X(w_2 - w_1) - y$$

Hence $h(v_{\text{opt}}) = -\frac{1}{2}||X(w_2 - w_1) - y||_2^2$. Writing the dual function gives:

$$g(w_1, w_2) = \inf_{v\in\mathbb{R}^n} L(v, w_1, w_2) = -\frac{1}{2}||X(w_2 - w_1) - y||_2^2 - \lambda\mathbb{1}_d^T(w_1 + w_2)$$

We derive a problem equivalent of the dual problem of the QP dual:

$$\begin{cases} \text{minimize} \\ w_1 \in \mathbb{R}^d, w_2 \in \mathbb{R}^d \\ \text{s.t. } w_1 \succcurlyeq 0, w_2 \succcurlyeq 0 \end{cases} \frac{1}{2} \quad ||X(w_2 - w_1) - y||_2^2 + \lambda||w_1 + w_2||_1$$

where we recognize a form similar to the LASSO.

We know from the relations on Lagrange multipliers that we have $\frac{-1}{t(Av^*-b)_i} = \begin{pmatrix} w_1^* \\ w_2^* \end{pmatrix}_i$ for a component $i$ of the vector considered. As a consequence, looking at $w^* = w_2^* - w_1^*$ means looking at the optimum point of the LASSO problem.

We generate a matrix $X$ with $d = 1200$ and $n = 8$. We generate a vector hidden_w of size $d$ having approximately 1% of non-zero coefficients. In order to comment the results on paper, we remove the randomness with np.random.seed(123). Of course, you can comment this line. We generate y by perturbing X.hidden_w by a normal random variable of standard deviation 0.1.

We use the notation LASSO$(w) = \frac{1}{2}||Xw - y||_2^2 + \lambda||w||_1$. Of course, LASSO$(w^*) <$ LASSO(hidden_w) because y is close to X.hidden_w but has been perturbed. Running an optimization allows to find a more appropriate w such that Xw is nearer to y than X.hidden_w.

For np.random.seed(123), the size of the support of the vector hidden_w is 18, whereas the size of the support of $w^*$ is 4. This is the experimental proof of the fact that LASSO enforces the sparsity of the solution. Indeed, the regularization by $\lambda||w||_1$ is low when the support of w is small.

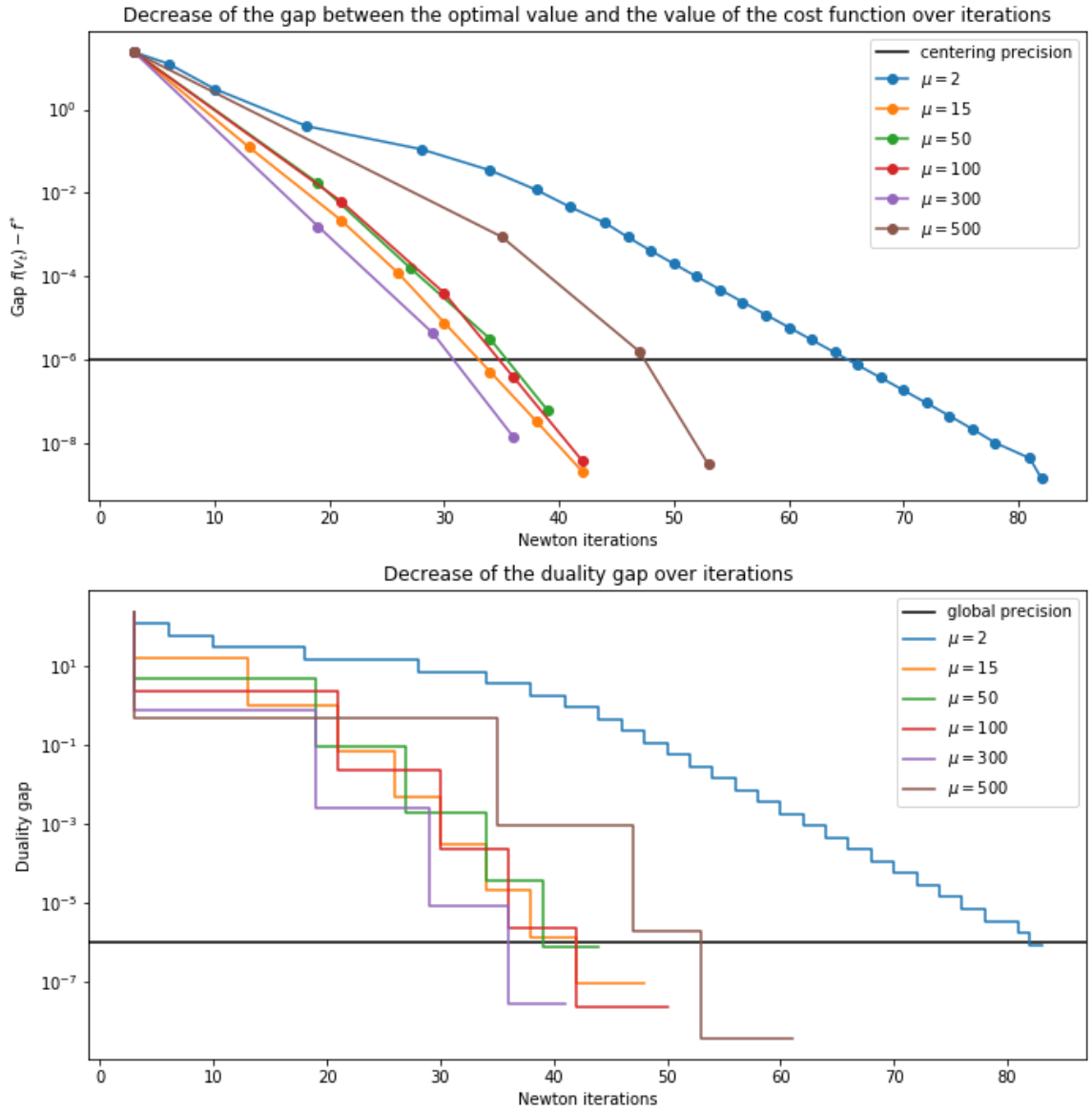


Figure 3: We plot the precision criterion and the gap $f(v_t) - f^*$ in semi-log scale (using the best value found for $f$ as a surrogate for $f^*$).