Convex Optimization - Homework 3 - MVA

LIU Vincent

November 17, 2020

1 LASSO to a quadratic problem

$$\min_{w} \qquad \frac{1}{2}\|Xw - y\|_2^2 + \lambda\|w\|_1 \tag{LASSO}$$

Where $w \in \mathbf{R}^d, X = (x_1^T, \dots, x_n^T)^T \in \mathbf{R}^{n\times d}, y = (y_1, \dots, y_n)^T \in \mathbf{R}^n, \lambda > 0$.

We introduce $z = Xw - y \in \mathbf{R}^n$ and reformulate (LASSO) as:

$$\begin{aligned} \min_{w,z} \qquad & \frac{1}{2}\|z\|_2^2 + \lambda\|w\|_1 \\ \text{subject to} \qquad & z = Xw - y. \end{aligned}$$

Then, we derive the dual function:

$$\begin{aligned} g(\nu) &= \inf_{w,z}(\frac{1}{2}\|z\|_2^2 + \lambda\|w\|_1 + \nu^T(Xw - y - z)) \\ &= \inf_{w,z}(\frac{1}{2}\|z\|_2^2 + \lambda\|w\|_1 + \nu^T Xw - \nu^T y - \nu^T z) \\ &= -v^T y + \inf_{w}(\lambda\|w\|_1 + (X^T\nu)^T w) + \inf_{z}(\frac{1}{2}\|z\|_2^2 - \nu^T z) \\ &= -\nu^T y + \inf_{w}(\lambda\|w\|_1 + (X^T\nu)^T w) + \inf_{z}(\frac{1}{2}\|z - \nu\|_2^2 - \frac{1}{2}\|\nu\|_2^2) \\ &= -\frac{1}{2}\nu^T\nu - \nu^T y + \inf_{w}(\lambda\|w\|_1 + (X^T\nu)^T w) \\ &= \begin{cases} -\frac{1}{2}\nu^T\nu - \nu^T y & \text{if } \|X^T\nu\|_\infty \le \lambda \\ -\infty & \text{o.w} \end{cases} \end{aligned}$$

Where $\nu \in \mathbf{R}^n$ is the vector containing the Lagrange multipliers associated with the equality constraint $z = Xw - y$.

We can write the Lagrange dual problem associated with (LASSO):

$$\begin{aligned} \max_{\nu} \qquad & -\frac{1}{2}\nu^T\nu - \nu^T y \\ \text{subject to} \qquad & \|X^T\nu\|_\infty \le \lambda \end{aligned}$$

as a minimization optimization problem:

$$\begin{aligned} \min_{\nu} \quad & \frac{1}{2}\nu^T\nu + \nu^T y \\ \text{subject to} \quad & \|X^T\nu\|_\infty \leq \lambda \end{aligned}$$

By reformulating the constraint:

$$\|X^T\nu\|_\infty \leq \lambda$$

As:

$$\max_{i=1,\dots,d} |(X^T)_i\nu| \leq \lambda$$

Which can be rewritten as:

$$|(X^T)_i\nu| \leq \lambda \quad \text{for every } i = 1, \dots, d$$

Which is equivalent to:

$$\begin{aligned} (X^T)_i\nu &\leq \lambda \quad \text{for every } i = 1, \dots, d \\ -(X^T)_i\nu &\leq \lambda \quad \text{for every } i = 1, \dots, d \end{aligned}$$

we can format the dual problem as a Quadratic Problem, that is:

$$\begin{aligned} \min_{\nu} \quad & \nu^T Q\nu + p^T\nu \\ \text{subject to} \quad & A\nu \leq b \end{aligned} \tag{QP}$$

Where $Q = \frac{I}{2} \in \mathbf{R}^{n\times n} \geq 0, p = y \in \mathbf{R}^n, A = \begin{bmatrix} X^T \\ -X^T \end{bmatrix} \in \mathbf{R}^{2d\times n}$ and b is the vector of $\mathbf{R}^{2d}$ with $b_i = \lambda$ for $i = 1, \dots, 2d$.

2 Implementation

2.1 Centering step

Given $t > 0$, find $v^*(t)$ that minimizes the following problem:

$$\min_{v} \quad f(v) \tag{CENTERING\_PATH}$$

where:

$$\begin{aligned} f(v) &= tf_0(v) + \phi(v) \\ &= tf_0(v) - \sum_{i=0}^{2d} log(-f_i(v)) \\ &= t(v^TQv + y^Tv) - \sum_{i=0}^{2d} \log(-A_iv + b_i) \end{aligned}$$

We write down the gradient of the objective function f:

$$\begin{aligned}\nabla f(v) &= t\nabla f_0(v) + \sum_{i=0}^{2d} \frac{1}{-f_i(v)} \nabla f_i(v) \\ &= t(2Qv + y) + \sum_{i=0}^{2d} \frac{1}{-A_i v + b_i} A_i^T \in \mathbf{R}^n\end{aligned}$$

as well as its Hessian:

$$\begin{aligned}\nabla^2 f(v) &= t\nabla^2 f_0(v) + \sum_{i=0}^{2d} \frac{1}{f_i(v)^2} \nabla f_i(v) f_i(v)^T + \sum_{i=0}^{2d} \frac{1}{-f_i(v)} \nabla^2 f_i(v) \\ &= 2tQ + \sum_{i=0}^{2d} \frac{1}{(-A_i v + b_i)^2} A_i^T A_i \in \mathbf{R}^{n \times n}\end{aligned}$$

The centering step consists of applying Newton method to solve (CENTERING_PATH). The implementation was written in Python and it involves manipulating numpy array objects. I followed the Newton's method pseudo code provided in slide 19/53 together with the baktracking line search 8/53 in the UnconstrainedEquality chapter. Some considerations while I implemented the methods are:

- I stored each of the $\frac{1}{-A_i v + b_i}$ in an array of shape (2d,) named *grad_phi_denom* so we can use it both in the gradient and the hessian computation.
- I stored the outer product $A_i^T A_i$ appearing in the computation of the hessian before the loop, in an array of shape (2d, n, n) named *grad_phi_num*, since we have just to compute it once.
- To compute the quantity $\sum_{i=0}^{2d} \frac{1}{-A_i v + b_i} A_i^T$ of the gradient, we can see it as a dot product between A and *grad_phi_denom*. Same goes for the computation of $\sum_{i=0}^{2d} \frac{1}{(-A_i v + b_i)^2} A_i^T A_i$ for the hessian but we use instead *np.tensordot* between *grad_phi_denom* at the power 2 and *grad_phi_num*.
- For the backtracking line search, we need to check if the solution is feasible before evaluating the logarithm (otherwise it is returning a NaN value). So we need to multiply t by β as long as the solution is not feasible, before doing the backtracking line search evaluation $f(x + t\Delta x) < f(x) + \alpha t \nabla f(x)^T \Delta x$.

2.2 Barrier method

Once we have implemented the centering step, the barrier method is straight forward. I followed the slide 13/36 of the BarrierMethod chapter.

2.3 Run the code

The code is in the LIU-Vincent-DM3-code.ipynb file. It contains the function backtracking_line_search, centering_step, barr_method as well as three utility functions is_feasible, f_0 and f. It contains also sanity checks, experiments and plots.

```
jupyter-notebook LIU-Vincent-DM3-code.ipynb
```

Listing 1: Run the code

3 Analysis

3.1 Duality gap

I ran the code with $d = 100$, $n = 50$, $\lambda = 10$, $t = 1$, $\epsilon = 1^{-6}$, $\alpha = 0.1$, $\beta = 0.6$, $v_0 = 0_{\mathbf{R}^n}$.The initial point v_0 is feasible since $A.0_{\mathbf{R}^n} \leq 10$.

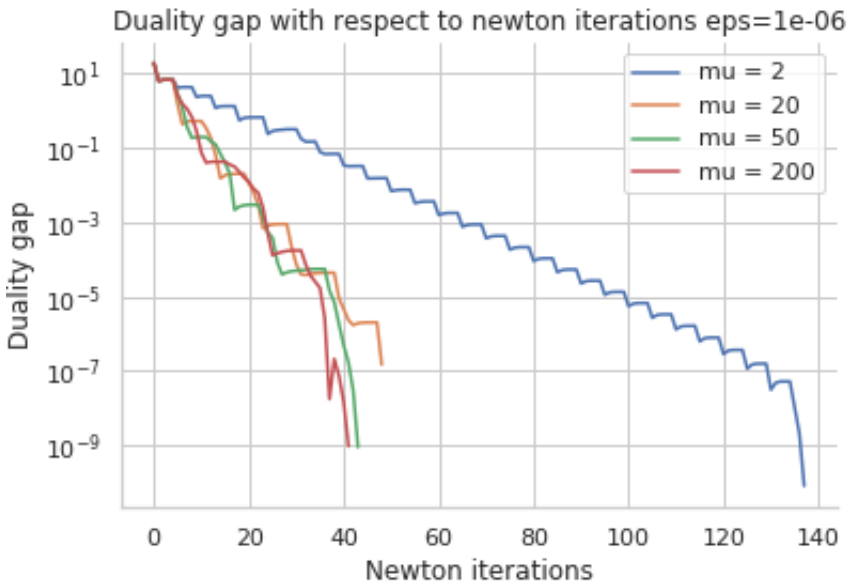

Figure 1: Duality gap w.r.t. Newton iterations (semi log y scale)

Figure 1 shows the duality gap with respect to Newton iterations for different values of μ. Each vertical step is an outer iteration whereas each horizontal step is an inner iteration. We have approximately linear convergence of our result. We observe that we do small steps when μ is small ($\mu = 2$) and we do bigger but fewer steps when μ is increased ($\mu = 20, 50, 200$).

3.2 Impact on primal solution w

The Lagrangian of the primal problem was:

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

At the optimal point, the KKT conditions are sufficient since the problem is convex (objective function is convex as sum of convex functions and constraints are linear).

We have first the stationary condition for non differential objective function w.r.t w :

$$0 \in \partial_w L(w^*, z^*, v^*) \iff 0 \in \lambda\partial_w(||w^*||_1) + (v^*)^T X \qquad \text{(KKT 1)}$$

with $\partial_w(||w^*||_1) \subset \mathbf{R}^d$. If $g \in \partial_w(||w^*||_1)$ then $g_i = sign(w_i^*)$ if $|w_i^*| > 0$ and $g_i \in [-1, 1]$ otherwise. We deduce that:

$$((v^*)^T X)_i = \begin{cases} \text{- } \lambda & \text{if } w_i^* > 0 \\ \text{+ } \lambda & \text{if } w_i^* < 0 \\ [\text{-}\lambda, \text{+}\lambda] & \text{if } w_i^* = 0 \end{cases}$$

The gradient of L w.r.t z vanishes:

$$\nabla_z L(w^*, z^*, v^*) = 0 \iff z^* - \nu^* = 0 \iff z^* = \nu^* \qquad \text{(KKT 2)}$$

The primal constraint is feasible:

$$z^* = Xw^* - y \qquad \text{(KKT 3)}$$

So is the dual constraint:

$$\|X^T v^*\|_\infty \leq \lambda \qquad \text{(KKT 4)}$$

Putting the two equations (KKT 2) and (KKT 3) together gives:

$$v^* + y = Xw^*$$

In order to recover the solution w^* of the primal problem, the first step is to set to 0 the values of w_i^* for which $|(v^*)^T X)_i| < \lambda$. Assuming that $\text{rank}(X) = n << d$ with high probability, the pseudo inverse $(X^T X)^{-1} X^T$is invertible, so we can then use the pseudo inverse to recover the remaining non zero coefficients:

$$\bar{w}^* = (\bar{X}^T \bar{X})^{-1} \bar{X}^T (v^* + y)$$

where $\bar{w}$ refers to the vector of non zero coefficient of w.

In the code, we check if these KKT conditions hold altogether. We can also check strong duality of our convex problem by evaluating the primal and dual function at respectively w^* and v^*. For our generated data, we have 7 non zero coefficients at the end and strong duality holds with value 18. This holds TRUE for every value of μ.

3.3 Influence of μ

Figure 2 shows the newton iterations with respect to μ. We see that the number of Newton iteration steps decreases as μ increases. For $\mu > 25$, we see that the number of iterations is not really sensitive of the value of μ.

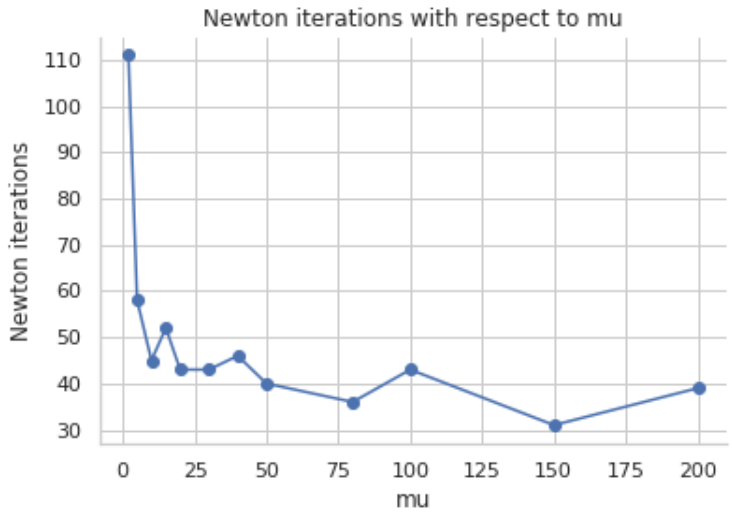

Figure 2: Newton iterations w.r.t μ

3.4 Visualize Newton decrement

Figure 3 shows Newton decrement w.r.t to newton inner iteration at a given outer iteration. We see that the Newton decrement quadratically decreases regardless of the value of μ.

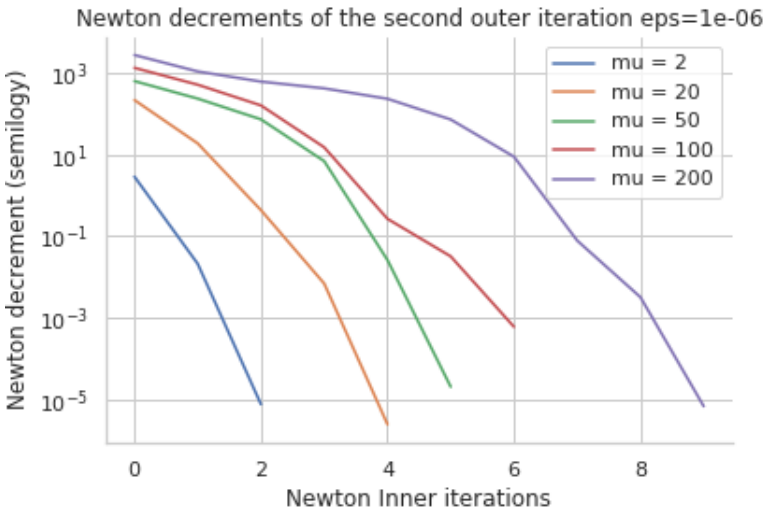

Figure 3: Newton decrement w.r.t to newton inner iteration for a fixed outer iteration, in semi log y scale