HW3

Gabriel ROMON

1

In order to turn the problem into a constrained one, we add an auxiliary variable $z \in \mathbb{R}^n$:

$$\min_{w,z} \frac{1}{2}\|z\|_2^2 + \lambda\|w\|_1 \quad \text{s.t} \quad z = Xw - y$$

The Lagrangian of this problem is $L\left(\begin{pmatrix} w \\ z \end{pmatrix}, \nu\right) = \frac{1}{2}\|z\|_2^2 + \lambda\|w\|_1 + \nu^T(z - Xw + y)$.
For a fixed ν, note that

$$\begin{aligned}
\inf_{w,z} L\left(\begin{pmatrix} w \\ z \end{pmatrix}, \nu\right) &= \inf_{w,z}\left[\frac{1}{2}\|z\|_2^2 + \lambda\|w\|_1 + \nu^T(z - Xw + y)\right] \\
&= \inf_{z}\left[\frac{1}{2}\|z\|_2^2 + \nu^T z + \inf_w(\lambda\|w\|_1 - \nu^T Xw)\right] + \nu^T y \\
&= \inf_{z}\left[\frac{1}{2}\|z\|_2^2 + \nu^T z - \lambda \sup_w(\frac{1}{\lambda}(X^T\nu)^T w - \|w\|_1)\right] + \nu^T y \\
&= \inf_{z}\left[\frac{1}{2}\|z\|_2^2 + \nu^T z - \lambda f^*(\frac{1}{\lambda}X^T\nu)\right] + \nu^T y
\end{aligned}$$

where f^* is the conjugate of the $\|\cdot\|_1$ function. f^* has already been computed in HW2 so we skip its derivation.

Hence

$$\inf_{w,z} L\left(\begin{pmatrix} w \\ z \end{pmatrix}, \nu\right) = \begin{cases} \inf_z\left[\frac{1}{2}\|z\|_2^2 + \nu^T z\right] + \nu^T y & \text{if} \quad \|X^T\nu\|_\infty \leq \lambda \\ -\infty & \text{otherwise} \end{cases}$$

$z \mapsto \frac{1}{2}\|z\|_2^2 + \nu^T z$ is a convex differentiable function, and is thus minimized at a critical point. We find that $z = -\nu$ is a critical point, hence $\inf_z\left[\frac{1}{2}\|z\|_2^2 + \nu^T z\right] = -\frac{1}{2}\|\nu\|_2^2$ and finally

$$\inf_{w,z} L\left(\begin{pmatrix} w \\ z \end{pmatrix}, \nu\right) = \begin{cases} -\frac{1}{2}\|\nu\|_2^2 + \nu^T y & \text{if} \quad \|X^T\nu\|_\infty \leq \lambda \\ -\infty & \text{otherwise} \end{cases}$$

The dual problem of LASSO is therefore

$$\max_\nu -\frac{1}{2}\|\nu\|_2^2 + \nu^T y \quad \text{s.t} \quad \|X^T\nu\|_\infty \leq \lambda$$

Since $\|X^T\nu\|_\infty \leq \lambda \iff \begin{pmatrix} X^T \\ -X^T \end{pmatrix}\nu \leq \lambda\mathbf{1}_{2d}$, the problem rewrites as

$$\max_\nu -\frac{1}{2}\nu^T I_n \nu + y^T\nu \quad \text{s.t} \quad \begin{pmatrix} X^T \\ -X^T \end{pmatrix}\nu \leq \lambda\mathbf{1}_{2d}$$

and it suffices to solve

$$\min_\nu \frac{1}{2}\nu^T I_n \nu - y^T\nu \quad \text{s.t} \quad \begin{pmatrix} X^T \\ -X^T \end{pmatrix}\nu \leq \lambda\mathbf{1}_{2d}$$

which turns into the Quadratic Problem

$$\min_\nu \nu^T Q\nu + p^T\nu \quad \text{s.t} \quad A\nu \leq \lambda b$$

with

$$Q = \frac{1}{2}I_n, \quad p = -y, \quad A = \begin{pmatrix} X^T \\ -X^T \end{pmatrix}, \quad b = \lambda\mathbf{1}_{2d}$$

2

Our implementation of the barrier method is the strict Python transcription of the algorithms in Boyd's book. For our case, it is important to note that there are no equality constraints, so we may use Newton's unconstrained minimization method for the centering steps.
Throughout the code, we let

$$f_t : \nu \mapsto t(\nu^T Q \nu + p^T) - \sum_{i=1}^{d} \log(b_i - L_i \nu)$$

where L_i is the i-th line of A. For the Newton step it is necessary to compute the gradient and Hessian of f_t, which are given by

$$\nabla f_t(\nu) = t((Q + Q^T)\nu + p) + \sum_{i=1}^{2d} \frac{1}{b_i - L_i \nu} L_i^T$$

$$H(f_t)(\nu) = 2tQ + \sum_{i=1}^{2d} \frac{1}{(b_i - L_i \nu)^2} L_i^T L_i$$

The step size in Newton's method is obtained by backtracking line search.

The parameters of the method can be decomposed as follows:

- in the line search: $\alpha \in \left(0, \frac{1}{2}\right)$ and $\beta \in (0, 1)$
- in Newton's method: a tolerance $\varepsilon_{\text{Newton}}$
- in the barrier method: an initial $t^{(0)} > 0$, a tolerance $\varepsilon_{\text{barrier}}$, and $\mu > 1$

3

We set arbitrary values of the aforementioned parameters and we work with dimensions $d = 50$, $n = 10$. The results of the method for a randomly generated X and y are displayed in Figure 1. A trade-off can be observed according to the value of μ. If μ is small ($\mu = 2$ in this example), there is a small number of Newton steps (inner iterations) per outer iteration, which is counterbalanced by a large number of outer iterations. When μ is large ($\mu = 300$ here), there are more inner iterations and less outer iterations.
A balance between the two is reached for $\mu = 50$, which corresponds to the green curve in Figure 1.

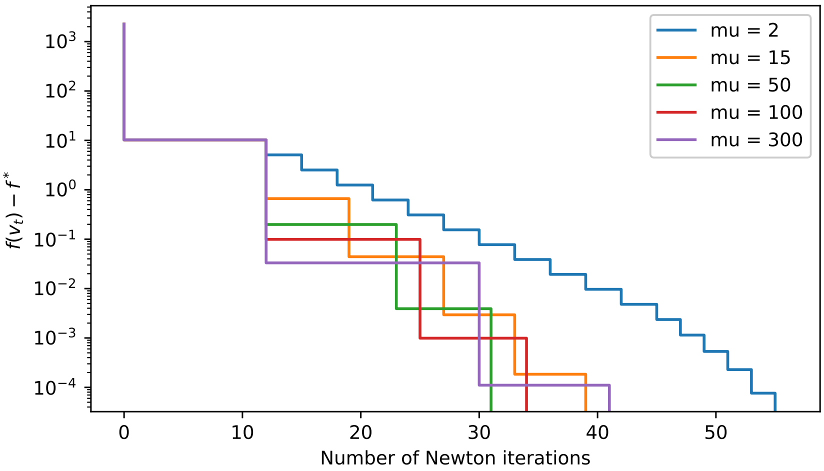

Figure 1: Progress of barrier method, showing approximate duality gap versus cumulative number of Newton steps, for different μ's