Non-smooth hyper-parameter learning

Clarice Poon

March 30, 2022

Consider

$$\min_y h(y), \quad \text{where} \quad h(y) = f(y, x(y)),$$
$$\text{s.t.} \quad x(y) \in \operatorname{argmin}_x g(y, x).$$

In general, we have

$$\partial_x g(y, x(y)) = 0,$$

and by the implicit function theorem, provided that $\partial_x^2 g(y, x(y))$ is invertible, $y \mapsto x(y)$ is differentiable with

$$x'(y) = -\partial_{xx} g(y, x(y))^{-1} \partial_{yx} g(y, x(y)).$$

One can then evaluate the gradient of h as

$$\nabla h(y) = \partial_y f(y, x(y)) - \partial_x f(y, x(y))^\top \partial_{xx} g(y, x(y))^{-1} \partial_{yx} g(y, x(y)). \tag{1}$$

Quasi-Newton? One approach is to find p_k such taht

$$\partial_{yx} g(y, x(y)) \approx \partial_{yy} g(y, x(y)) p_k$$

and evaluate $\nabla h(y) \approx \partial_y f(y, x(y)) - \partial_x f(y, x(y))^\top p_k$.

Can we do a quasi-Newton approach? We know that

$$\partial_x g(y, x(y)) = 0$$

Let $x_k \triangleq x(y_k)$. Then

$$\begin{aligned}
\partial_x g(y_k, x(y_{k-1})) &= \partial_x g(y_k, x_{k-1}) - \partial_x g(y_k, x_k) \\
&= \partial_{xx} g(y_k, x_k)(x_{k-1} - x_k) + o(\|x_{k-1} - x_k\|)
\end{aligned}$$

Suppose we find B_k such that it minimises

$$\min_B \|B \partial_x g(y_k, x(y_{k-1})) - (x_{k-1} - x_k)\|$$

and treat it as an approximation to $\partial_{xx} g(y_k, x(y_k))^{-1}$. The idea is to compute

$$\nabla h(y_k) \approx \partial_y f(y_k, x(y_k)) - \partial_x f(y_k, x(y_k))^\top B_k \partial_{yx} g(y_k, x(y_k)).$$

One possible update of B_k is as $\tau_k \mathrm{Id} + u_k u_k^\top$. Define $s_k = x_{k-1} - x_k$ and $z_k = \partial_y g(y_k, x(y_{k-1}))$. We want to find diagonal + rank-1 matrix B to minimise

$$\min_B \|Bz_k - s_k\|$$

i) Define $\tau_k = \langle s_k,\, z_k\rangle / \|z_k\|^2$ and project onto $[\tau_{\min}, \tau_{\max}]$. Note that before projection, $\tau_k = \mathrm{argmin}_\tau \|\tau z_k - s_k\|$.

ii) Let $B_0 \triangleq \gamma\tau_k \mathrm{Id}$ where $\gamma \in (0,1)$.

iii) If $\langle s_k - B_0 z_k,\, z_k\rangle \leqslant 10^{-8}\|z_k\|\|s - B_0 z_k\|$ the $U_k = 0$. Else:

$$U_k = \frac{(s_k - B_0 z_k)(s_k - B_0 z_k)^\top}{\langle s_k - B_0 z_k,\, z_k\rangle}.$$

iv) Let $B_k = B_0 + U_k$.

Note that for step iii) the choice of U_k is precisely finding $U_k = uu^\top$ such that

$$B_0 z_k + u\langle u,\, z_k\rangle - s_k = 0.$$

[ToDo:

1. **If we repeatedly updated B_k with rank-1 matrices, show that B_k converges to $\partial_{xx} g(y_*, x(y_*))^{-1}$.**

2. **Suppose that $f = g$ and at iteration k, we have an approximate solution $\hat{x}_k \approx x(y_k)$. Possible ways of computing $\nabla h(y_k)$ include**

   i) $p_1 = \partial_y f(y_k, \hat{x}_k))$

   ii) $p_2 = \partial_y f(y_k, \hat{x}_k)) - \partial_x f(y_k, \hat{x}_k)^\top B_k \partial_{yx} f(y_k, \hat{x}_k)$

   iii) $p_3 = \partial_y f(y_k, \hat{x}_k) + \partial_x f(y_k, \hat{x}_k)^\top \partial_y \hat{x}_k$ **where we obtain** $\partial_y \hat{x}_k$ **via autodiff.**

   The first and 3rd option have been analysed recently (Ablin et al). For the second approach, can we bound the difference between taking approximation B_k and the true Hessian?

]

A differentiable approach to nonsmooth bilevel programming One example is where g is nonsmooth is when y correspond to a hyperparamter λ and x is the lasso regression coefficients:

$$f(\lambda, \beta) \triangleq \|A_{\text{test}}\beta - y\|^2 \quad \text{and} \quad g(\lambda, \beta) \triangleq \frac{1}{2}\|A_{\text{train}}\beta - b\|_2^2 + \lambda\|\beta\|_1.$$

The difficulty is in this case is that since g is non-smooth, the formula (1) cannot be used. One alternative is to consider instead

$$f(\lambda, (u,v)) \triangleq \|A_{\text{test}} uv - y\|^2 \quad \text{and} \quad g(\lambda, (u,v)) \triangleq \|A_{\text{train}} uv - b\|_2^2 + \lambda\|u\|^2/2 + \lambda\|v\|^2/2.$$

The advantage with this approach is that g is a smooth function and one can show that the Hessian of g is invertible when $\beta \triangleq u(\lambda) \odot v(\lambda)$ is a nondegenerate solution, that is,

$$\max_{i \notin \mathrm{Supp}(\beta)} |A_{\mathrm{train}}^\top (A_{\mathrm{train}}\beta - b)|_i < 1.$$

Things to do

- Check properties of the Hessian of g.
- Acceleration using support pruning.

The square root lasso The square root lasso is

$$\min_{\beta \in \mathbb{R}^n} \|X\beta - y\|_2 + \lambda \|\beta\|_1.$$

One interesting aspect of this is that when $y = X\beta_0 + w$, the minimiser β satisfies

$$\|\beta - \beta_0\| \lesssim \|w\|$$

for some constant λ. This is remarkable since, for the Lasso, to achieve this kind of error bound, one would require that $\lambda \sim \|w\|$ and some knowledge of the noise level is required.

One remark is that the square root lasso is equivalent to

$$\min_{\sigma > 0} \min_{\beta} \frac{1}{2\sigma} \|X\beta - y\|_2^2 + \frac{\sigma}{2} + \lambda \|\beta\|_1,$$

and we can therefore write this in the bilevel formulation with

$$f(\sigma, \beta) = \frac{1}{2\sigma} \|X\beta - y\|_2^2 + \frac{\sigma}{2} + \lambda \|\beta\|_1$$

and

$$g(\sigma, \beta) = \frac{1}{2\sigma} \|X\beta - y\|_2^2 + \lambda \|\beta\|_1.$$

One question I have is what happens when we consider

$$\begin{aligned} f(\sigma, \beta) &= \frac{1}{2\sigma} \|A_{\mathrm{test}}\beta - y\|_2^2 + \frac{\varepsilon\sigma}{2} + \iota_{\sigma > 0} \\ g(\sigma, \beta) &= \frac{1}{2\sigma} \|A_{\mathrm{train}}\beta - y\|_2^2 + \|\beta\|_1 \end{aligned}$$

This is precisely the hyperparameter learning framework but with added regularisation on the parameter σ. Note that the outer problem can be written as an unconstrained smooth problem as follows: Let $z = \sqrt{\sigma}$ and $v = A_{\mathrm{test}}\beta - y$, then

$$\min_{z \in \mathbb{R}} f(z, \beta(z^2)), \quad \text{where} \quad f(z, \beta) \triangleq \min_{zv = A_{\mathrm{test}}\beta - y} \frac{1}{2}\|v\|^2 + \frac{\varepsilon}{2} z^2,$$
$$\beta(z^2) \triangleq \mathrm{argmin}_\beta\, g(z^2, \beta)$$

Notice that the minimisation problem in f is convex wrt v, so by taking the convex dual,

$$f(z,\beta) = \max_{\alpha\in\mathbb{R}^m} -\frac{\varepsilon}{2} z^2 \|\alpha\|^2 + \frac{\varepsilon}{2} z^2 + \langle \alpha,\, -A_{\mathrm{test}}\beta + y\rangle$$

The maximiser α is unique (as the problem is strongly concave) and

$$\partial_\beta f = -A_{\mathrm{test}}^\top \alpha \quad \text{and} \quad \partial_z f = z\|\alpha\|^2.$$

Numerically, we can certainly handle this, the question is whether this kind of regularisation is interesting in practice.

Let $F(\sigma) = f(\sigma, \beta(\sigma))$. Let's look at the optimality conditions

$$\partial_\sigma f = \frac{-1}{\sigma^2}\|A_{\mathrm{test}}\beta - y_{\mathrm{test}}\|^2 + \frac{\varepsilon}{2}$$
$$\partial_\beta f = \frac{1}{\sigma} A_{\mathrm{test}}^\top (A_{\mathrm{test}}\beta - y_{\mathrm{test}})$$

Also, $\beta = \beta(\sigma)$ satisfies

$$A_{\mathrm{train}}^\top A_{\mathrm{train}}\beta = A_{\mathrm{train}}^\top y - \sigma\,\mathrm{sign}(\beta)$$

In general, $\sigma \mapsto \beta(\sigma)$ is differentiable almost everywhere with gradient

$$\beta'(\sigma) = -(A_{\mathrm{train}}^\top A_{\mathrm{train}})_{J,J}^{-1}\,\mathrm{sign}(\beta).$$

where $J = \mathrm{Supp}(\beta)$. So, when $F'(\sigma) = 0$, we have

$$\frac{-1}{\sigma^2}\|A_{\mathrm{test}}\beta - y_{\mathrm{test}}\|^2 - \frac{1}{\sigma}\langle (A_{\mathrm{test}}\beta - y_{\mathrm{test}}),\, A_{\mathrm{test}}(A_{\mathrm{train}}^\top A_{\mathrm{train}})_{J,J}^{-1}\,\mathrm{sign}(\beta)\rangle + \frac{\varepsilon}{2} = 0$$

which implies $\lambda = 1/\sigma$ satisfies, for $C \triangleq \langle (A_{\mathrm{test}}\beta - y_{\mathrm{test}}),\, A_{\mathrm{test}}(A_{\mathrm{train}}^\top A_{\mathrm{train}})_{J,J}^{-1}\,\mathrm{sign}(\beta)\rangle$,

$$\lambda = \frac{-C + \sqrt{C^2 + 2\varepsilon\|A_{\mathrm{test}}\beta - y_{\mathrm{test}}\|^2}}{2\|A_{\mathrm{test}}\beta - y_{\mathrm{test}}\|^2}$$

NB: For the standard problem where $f(\sigma,\beta) = \frac{1}{2}\|A_{\mathrm{test}}\beta - y\|^2$, then

$$F'(\sigma) = -\langle (A_{\mathrm{test}}\beta - y_{\mathrm{test}}),\, A_{\mathrm{test}}(A_{\mathrm{train}}^\top A_{\mathrm{train}})_{J,J}^{-1}\,\mathrm{sign}(\beta)\rangle.$$

Suppose $A_{\mathrm{test}} = A_{\mathrm{train}}$, then this says that

$$F'(\sigma) = \sigma\langle (A_{\mathrm{train}}^\top A_{\mathrm{train}})_{J,J}^{-1}\,\mathrm{sign}(\beta),\, \mathrm{sign}(\beta)\rangle > 0$$

which means that we optimise to $\sigma = 0$ as expected.