Proximal Methods

Ran Wang

January 30, 2021

Chapter 1

Proximal Algorithms

1.1 Motivation

In this chapter, we briefly outline the problem of minimizing functions that are not necessarily differentiable. A typical example is the l_1-regularized problem. For example, the object might look like

$$\min_{\beta} \sum_{i=1}^{N} (y_i - x_i^t \beta)^2 + \lambda \|\beta\|_1.$$

Here, β is the parameter we want to find. Should we not have $\lambda\|\beta\|_1$, then everything is differentiable and can be solved use quasi-Newton methods, among other things. However, the absolute value function is not differentiable everywhere, which causes problems.

The first solution is by consider *sub-differentials*.Sub-differentials are defined as

$$\partial f(x) = \left\{ y \mid f(z) \geq f(x) + y^T(z - x) \text{ for all } z \in \operatorname{dom} f \right\},$$

where $\operatorname{dom} f$ is the domain of the function. Note that if a function is differentaible then $\partial f = \{\nabla f\}$. However, in general case, the sub-sdifferential is not a singleton.

For simplicity, we assume all the functions we discuss are subdifferentiable.

1.2 Proximal Algorithms

The proximal operator is defined as

$$\operatorname{prox}_f(v) = \operatorname*{argmin}_x \left(f(x) + (1/2)\|x - v\|_2^2 \right).$$

As simple as the definition might look like, it has quite some nice results. The first one is a fixed-point properties. That is, the point $x^\star$ minimizes f if and only if

$$x^\star = \operatorname{prox}_f(x^\star).$$

Proof. First we show that if x^* is the minimizer, then $x^\star = \operatorname{prox}_f(x^\star)$. Note that for any x,

$$f(x) + (1/2)\left\|x - x^\star\right\|_2^2 \geq f(x^\star) = f(x^\star) + (1/2)\left\|x^\star - x^\star\right\|_2^2,$$

and thus by definition, $x^\star = \operatorname{prox}_f(x^\star)$.

Now consider the reverse case, let $\tilde{x} = \operatorname{prox}_f(v)$. Take the subdifferential operator, we see that this is equivalent to

$$0 \in \partial f(\tilde{x}) + (\tilde{x} - v).$$

Taking $\tilde{x} = v = x^\star$, it follows that $0 \in \partial f(x^\star)$, so $x^\star$ minimizes f. □

The second interesting, and rather surprising fact is that, the proximal operator is actually the resolvent of subdifferetial operator. More specifically,

$$\operatorname{prox}_{\lambda f} = (I + \lambda\partial f)^{-1}.$$

Proof. If $z \in (I + \lambda\partial f)^{-1}(x)$, then $0 \in \partial f(z) + (1/\lambda)(z - x)$. This implies $0 \in \partial_z\left(f(z) + (1/2\lambda)\|z - x\|_2^2\right)$. Now, since we can prove that $f(z) + (1/2\lambda)\|z - x\|_2^2$ is strongly convex, we can deduce that $z = \underset{u}{\operatorname{argmin}}\left(f(u) + (1/2\lambda)\|u - x\|_2^2\right)$. □

Finally, let us look at the case of minimzing $f + g$ where f is differentiable and g is not. A famous algorithms goes,

$$x^{k+1} := \operatorname{prox}_{\lambda^k g}\left(x^k - \lambda^k \nabla f\left(x^k\right)\right).$$

To see why, consider the fixed point version that is $x^* = \operatorname{prox}_{\lambda g}(x^* - \lambda\nabla f(x^*))$, we show that if this is true then x is indeed the solution to

$$x^* = \operatorname{argmin}_x f(x) + g(x).$$

Proof. Note that x^* is the minimzer if and only if $0 \in \nabla f(x^\star) + \partial g(x^\star)$. Now with some straight forward computation,

$$\begin{aligned}
0 &\in \lambda\nabla f(x^\star) + \lambda\partial g(x^\star) \\
0 &\in \lambda\nabla f(x^\star) - x^\star + x^\star + \lambda\partial g(x^\star) \\
(I + \lambda\partial g)(x^\star) &\ni (I - \lambda\nabla f)(x^\star) \\
x^\star &= (I + \lambda\partial g)^{-1}(I - \lambda\nabla f)(x^\star) \\
x^\star &= \operatorname{prox}_{\lambda g}(x^\star - \lambda\nabla f(x^\star))
\end{aligned}$$

□

1.3 Applications

Now let us return to the previous topics. Let us assume that the **negative** log-likelihood function is $f_\beta(X)$, here β is the parameter we want to estimate and X is the data, which include both the dependent and independent variables. The goal now is the minimize the following quantities.

$$f_\beta(X) + \lambda\|\beta\|_1,$$

where $\lambda > 0$ is a positive hyperparemeter, $\|\cdot\|$ is the l_1 norm, namely for any $x \in R^N$, $\|x\|_1 = \sum_{i=1}^N |x_i|$. Since the latter term is not differentiable at 0, we are in position to workout the solution as is indicated in the previous section.

Now let $w := \beta^k - \lambda^k \nabla f_{\beta^k}(X)$, where λ^k is the step-size, and β^k is the value of β at step k, and let $g : x \mapsto \lambda\|x\|_1$, we have

$$\begin{aligned}
\beta^{k+1} &:= \operatorname{prox}_{\lambda^k g}(w) \\
&= \operatorname{argmin}_x \left(\lambda^k g(x) + \frac{1}{2}\|x - w\|_2^2 \right) \\
&= \operatorname{argmin}_x \left(\lambda\|x\|_1 + \frac{1}{2\lambda_k}\|x - w\|_2^2 \right)
\end{aligned}$$

Now it suffices to calculate the for each i, since none of i depends on others. For that, note that (**exercises!**) we have

$$\underset{z}{\operatorname{argmin}} \frac{1}{2}\|\beta - z\|_2^2 + \lambda t\|z\|_1 = S_{\lambda t}(\beta)$$

where we have

$$[S_\lambda(\beta)]_i = \begin{cases} \beta_i - \lambda & \text{if } \beta_i > \lambda \\ 0 & \text{if } -\lambda \le \beta_i \le \lambda \\ \beta_i + \lambda & \text{if } \beta_i < -\lambda \end{cases}$$

Therefore we have

$$\left[\beta^{k+1}\right]_i = \begin{cases} w_i - \lambda\lambda_k & \text{if } w_i > \lambda\lambda_k \\ 0 & \text{if } -\lambda\lambda_k \le w_i \le \lambda\lambda_k \\ w_i + \lambda\lambda & \text{if } w_i < -\lambda\lambda_k \end{cases}$$

With that, one can easily implement the algorithm.