proximal gradient descent and newton method

Jieun Shin

May 24, 2023

1 Proximal gradient descent

Recall that proximal gradient descent operates on a problem

$$\min_x g(x) + h(x),$$

where g is convex, sooth and h isconvex. We choose initial $x^{(0)}$ and repeat for $k = 1, 2, 3, \ldots$

$$x^{(k)} = \text{prox}_t(x^{(k-1)} - t_k \nabla g(x^{(k-1)}))$$

where $\text{prox}_t(\cdot)$ is the proximal operator associated with h,

$$\text{prox}_t(x) = \underset{z}{\text{argmin}} \frac{1}{2t}||x - z||_2^2 + h(z)$$

- Difficulty of iterations is in applying prox, which only depends on h (assuming that ∇g is computable)
- Proximal gradient descent enjoys same convergence rate as its fully smooth version, hence useful when prox is efficient

Recall the motivation for proximal gradient: iteratively minimize a quadratic expansion in g, plus original h

$$\begin{aligned} x^+ &= \underset{z}{\text{argmin}} \frac{1}{2t}||x - t\nabla g(x) - z||_2^2 + h(z) \\ &= \underset{z}{\text{argmin}} \nabla g(x)^T(z - x) + \frac{1}{2t}||x - z||_2^2 + h(z) \end{aligned}$$

The quadratic approximation here uses Hessian equal to (scaled version of) the identity $\frac{1}{t}I$.

A fundamental difference between gradient descent and Newton's method was that the latter also iteratively minimized quadratic approximations, but these used the local Hessian of the function in question.

so what happens if we replace $\frac{1}{t}I$ in the above with $\nabla^2 g(x)$?

2 Proximal Newton method

This leads us to the proximal Newton method. Now we must define

$$\text{prox}_H(x) = \underset{z}{\text{argmin}} \frac{1}{2}||x - z||_H^2 + h(z)$$

where $||x||_H^2 = x^T H x$ defines a norm, given a matrix $H > 0$. This is a scaled proximal mapping. With $H = \frac{1}{t}I$, we get back to the proximal gradient method.

Starting with $x^{(0)}$, we repeat for $k = 1, 2, 3, ...$

$$\begin{aligned} y^{(k)} &= \text{prox}_{H_{k-1}}(x^{(k-1)} - H_{k-1}^{-1} \nabla g(x^{(k-1)})) \\ x^{(k)} &= x^{(k-1)} + t_k(y^{(k)} - x^{(k-1)}) \end{aligned}$$

Here $H_{k-1} = \nabla^2 g(x^{(k-1)})$, and t_k is a step size, which we choose by backtracking line search.