Lecture 7 : Momentum Methods, Max Functions

EE227C . Lecturer: Professor Martin Wainwright . Scribe: Alvin Wan

We will now consider momentum methods.

1. Heavy Ball: $x^{l+1} = x^l - \alpha \nabla f x(^l) + \beta(x^l - x^{l-1})$
2. Nesterov: $x^{l+1} = x^l - \alpha \nabla f(x^l - x^{l-1}) + \beta(x^l - x^{l-1})$

Heavy ball has the following runtime. $O(\sqrt{\frac{M}{m}} \log(\frac{1}{\epsilon}))$ where $\kappa = \frac{M}{m} \geq 1$ and $\sqrt{\kappa} \leq \kappa$, so it can perform poorly. We use Nesterov's accelerated gradient in practice, which has runtime $O(\sqrt{\frac{M}{m}} \log(\frac{1}{\epsilon}))$. Note this is not a descent method. It is possible that your iterates can oscillate up and down. Overall, it will decrease, but it has a sense of instability. This is an "optimal algorithm". This m-convex M-smooth function is not a contraction. This is proved by Wilson, Wibisono, Jordan (2016).

Apply original algorithm to $f_\mu(x) = g(x) + \frac{\mu}{2}\|x\|_2^2$, where $T(\epsilon) = O(\sqrt{\frac{\beta}{\epsilon}} \log(\cdots))$ and $T_{ord}(\epsilon) = O(\frac{\beta}{\epsilon})$. The essential result is that if you can accelerate strongly convex problems, you can also accelerate weakly convex problems.

1 Subgradients

Recall the subdifferential is a set. Take weak subgradient calculus. It is weak in the sense that it is not giving us a unique characterization of all elements in the subdifferential, $g \in \partial f(x)$. With strong subgradient calculus, we wish to characterize all elements in the set, $\partial f(x)$. The latter is otherwise known as Danskin's theorem.

Consider some subtleties. When is it true that $\partial(f_1 + f_2)(x) = \partial f_1(x) + \partial f_2(x)$, i.e., when $\partial(f_1 + f_2)(x) = \{g_1 + g_2 | g_j \in \partial f(x), j = 1, 2\}$. If $\text{dom}(f_1) = \text{dom}(f_2) = \mathbb{R}^d$ always true. Domain of a convex function $\text{dom}(f) = \{x \in \mathbb{R}^d | f(x) < +\infty\}$. When we consider ordinary convex functions, we assume the domain is all reals.

With extended-reals convex functions, we consider $f : \mathbb{R}^d : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$. We can define the following function, where $\mathrm{dom}(\mathbb{1}) = C$

$$\mathbb{1}_C = \begin{cases} 0 & \text{if } x \in C \\ +\infty & \text{otherwise} \end{cases}$$

This algorithm allows us to blur the lines between a constrained and unconstrained problem. We note the following

$$\min_{x \in C} f(x) \leftrightarrow \min_{x \in \mathbb{R}^d} \{f(x) + \mathbb{1}_C(x)\}$$

What's interesting is now we can throw unconstrained techniques at it. For convex h:

$$O \in \partial h(x^*) \leftrightarrow x^* \text{ is a minimizer} \leftrightarrow \langle \nabla f(x^*), z - x^* \rangle \geq 0, \forall z \in C\}$$

If $\mathrm{int}(\mathrm{dom} f_1 \cap \mathrm{dom} f_2) \neq \phi$ then $\partial(f_1 + f_2) = \partial f_1 + \partial f_2$. Keep in mind slater's condition.

2 Max Functions

These functions are of the following structure, where f_j is convex and differentiable:

$$f(x) = \max_{j=1,\ldots,N} f_j(x)$$

These functions may be piece-wise linear, $f(x) = \max_{j=1,\ldots,N}\{\langle a_j, x \rangle + b_j\}$ such as $f(x) = \max_{z \in Z} \phi(x, z)$.

We can write $f(x) = \max_{z \in [0,1]^N, \sum_{j=1}^N z_j = 1} \sum_{j=1}^N z_j f_j(x)$.

$$\partial f(x) = \mathrm{conv}\{\nabla f_j(x) | j \text{ s.t. } f_j(x) = f(x)\}$$

This is a strong rule for a max function.

Example: We're given to closed, convex sets $C_1, C_2 \subset \mathbb{R}^d$. Find $x^* \in C_1 \cup C_2\}$ assumed to be non-empty. We take the feasibility problem $y^{\ell+1} = \Pi_{C_1}(x^\ell), x^{\ell+1}\Pi_{C_2}(y^{\ell+1})$ and convert it into an optimization problem.

Give function $f : \mathbb{R}^d \to \mathbb{R}$ s.t. $f(x) = 0$ if $x \in C_1 \cup C_2$ and $f(x) > 0$ otherwise. Let us examine the following function.

$$f(x) = \max_{j=1,2,\ldots} \{\min_{y \in C_j} \|x - y\|_2\}$$

We are looking at the worst case of the L2 distances from your set. Note that the optimal value for that minimization problem is $\|x - \Pi_{C_j}(x)\|_2$. It gives us the properties we desire and is reasonably well behaved. First, we need to verify it's convex and second, we need to draw a weak rule from it. If we use the correct step size, we will get exactly the algorithm above.