First-Order Methods

Wednesday, January 24, 2018 12:44 PM

Call a method is a first-order Method if each iteration it uses all previous computed gradients $\nabla f(x_1), \nabla f(x_2), \ldots, \nabla f(x_{k-1})$ and initial x_1, $f(x_1), f(x_2), \ldots, f(x_{k-1})$ to generate a new point x_k.

In short, first-order Method can only access to f and ∇f.

Remark: There are other access model such as f^* and ∇f^*.

Usually, first-order method satisfies

$$x_k \in \text{span}(x_1, \nabla f(x_1), \nabla f(x_2), \ldots, \nabla f(x_{k-1}))$$

According to the definition, cutting plane methods are first-order.

However, the common use of the term "first-order method" refers to methods that only use "simple formula"

A typical first-order method:

$$x_{k+1} = x_k - \frac{1}{\beta} \nabla f(x_k) \quad \text{(gradient descent)}$$

Common belief: first-order method (those with simple formula) works well for large-scale Problems

My belief: There should be something that is better.

Practicality of Cutting Plane Methods.

How is it possible to say compute the center of gravity of 1 billions dimension convex sets is the runtime is $O(n^5)$?

→ I am writing a package with running time closer to linear. Let see how long does it take.

→ You never need to do so.

Note that all cutting plane method we mentioned satisfies

$$x_k = \text{span}(x_1, \nabla f(x_1), \nabla f(x_2), \ldots, \nabla f(x_{k-1})) \quad \text{(k-1 dim)}$$

Therefore, wlog, we only need to solve some k-1 dim problem at k^{th} step.

Therefore, wlog, we only need to solve some $k-1$ dim problem at k^{th} step.

Roughly speaking, if the cutting plane method involves solving a problem that takes $f(k)$ time for $\mathbb{R}^k$.

Then, the total time for the k^{th} iter is

$$\underbrace{T(\nabla f)}_{\text{compute gradient}} + \underbrace{nk}_{\text{maintain } k\text{-dim subspace}} + \underbrace{f(k)}_{\text{solve some easy problem}}$$

As I mentioned last time, for problem of the form $\sum f_i(a_i^T x - b_i)$, $T(\nabla f)$ is often dominated by computing Av for some v where $A = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}$

So, a more fair runtime estimate for cutting plane method is

$$\underbrace{nnz(A)}_{\text{number of nonzero}} + nk + f(k)$$

As comparison, L-BFGS takes $nnz(A) + nk$.

If it is a large-scale problem, $f(k) \ll nk$.

My beliefs: Cutting plane methods becomes more competitive as $n \to \infty$.

If the problem is large scale and we spent lots of time to get ∇f, Then it sounds silly to just combine these gradient by simple combination.

Note that IPM, CHOL, Multigrid, AGD, they all takes decades to become recognized. Not sure about cutting plane methods.

Thesis Question: Develope cutting-plane method that take advantage of $\sum f_i$ structure and Implement it.

Summary of first-order methods

Algorithm	errors (Assume the function is α strongly convex, β smooth 1-Lipschitz, $R = \|x_0 - x^*\|_2$ ⤵ equivalent		
Gradient Descent	$\frac{\beta R^2}{t}$ $\beta R^2 (1 - \frac{\alpha}{\beta})^t$		
Mirror Descent	$\frac{GR\sqrt{2}}{\sqrt{t}}$ $\frac{G^2}{2\alpha t}$	tight tight	⤵ reduction
Accelerated	$\frac{\beta R^2}{t^2}$	tight	⤵ "

Accelerated Gradient Descent	$G/\alpha t$	Tight
	$\beta R^2/t^2$	tight
	$\beta R^2(1-\sqrt{\frac{\alpha}{\beta}})^t$	$\tilde{O}$(tight)
Certain Cutting Plane Methods	$(1-\frac{1}{n})^{2n(t)}$	tight
	$\beta R^2(1-\sqrt{\frac{\alpha}{\beta}})^t$	$\tilde{O}$(tight)

Misconception: linear convergence ($(1-\varepsilon)^t$) is better than sublinear convergence ($\frac{1}{t^c}$)

We can always solve convex problem with linear convergence (aka $(1-\frac{1}{n})^{2n(t)}$)

Misconception: dimension independent result is faster for large-scale problem.

Even for very simple problem like $\sum(x_i-x_{i+1})^2$, we have $\frac{\beta}{\alpha}=n^2$.

Therefore dimension independent result can depends on dimension.

Relations between those bounds

(1) Suppose we have a way to find x s.t.

$$f(x_t)-f(x^*)\le\frac{G^2}{\alpha t}$$

for any α-strongly convex, G-Lip f

For any G-Lip convex f, we consider $f_\alpha(x)=f(x)+\frac{\alpha}{2}\|x-x_0\|^2$

We have that

$$f(x_t)-f(x^*)\le f_\alpha(x_t)-f_\alpha(x^*)+\frac{\alpha}{2}\|x^*-x_0\|^2$$

($\|x^*-x_0\|^2\le R^2$)

$$\le\frac{G^2}{\alpha t}+\frac{\alpha}{2}R^2$$ (used the algo above)

Choosing the best α, we have $f(x_t)-f(x^*)\le\frac{GR}{\sqrt{2}t}$

Here, if we have an algorithm achieves $\frac{G^2}{\alpha t}$,
we can turn it to an algorithm achieves $\frac{GR}{\sqrt{2}t}$.

(2) Suppose we know $\frac{\beta R^2}{t^2}$. If f is α-strongly convex, we have

② Suppose we know $\frac{BR^2}{t^2}$. If f is α-strongly convex, we have

$$\frac{\alpha}{2}\|x_t - x^*\|^2 \le f(x_t) - f(x^*) \le \frac{\beta}{t^2}\|x_0 - x^*\|^2$$

Set $t = 2\sqrt{\frac{\beta}{\alpha}}$, we have $\|x_t - x^*\|^2 \le \frac{1}{2}\|x_0 - x^*\|^2$

Hence, every $2\sqrt{\frac{\beta}{\alpha}}$, the distance to optimum decrease by $\frac{1}{2}$.

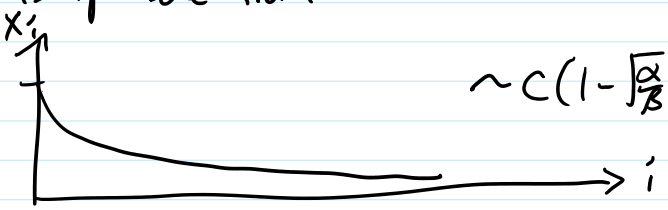
By restarting the algorithm, we have $BR^2(1 - \frac{1}{2}\sqrt{\frac{\alpha}{\beta}})^t$

Lower bounds

Consider the problem:

$$-x_1 + \frac{\beta}{2}\sum_{i=1}^{n}(x_i - x_{i+1})^2 + \frac{\alpha}{2}\sum_{i=1}^{n} x_i^2$$

The soln is of the from



Any approximate soln has support at least $\sqrt{\frac{\beta}{\alpha}}$.

Next, we that if

$$x_k = \text{span}(x_1, \nabla f(x_1), \nabla f(x_2), \ldots, \nabla f(x_{k-1}))$$

and if $x_1 = (1, 0, 0, \ldots, 0)$

Then x_k has only k non-zeros in the beginning.

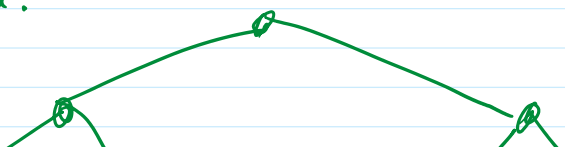
Therefore, we need $\sqrt{\frac{\beta}{\alpha}}$ iterations.

All first-order methods including cutting plane methods has this serious issue

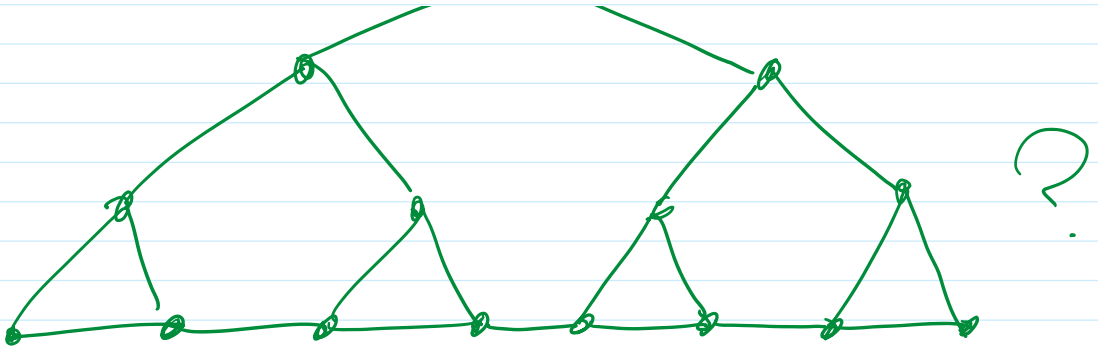
iter $\ge$ "diameter of the problem".

Open problem: Is there any way to help the connectivity for first order method?

How about



How about



?