Lecture 2

Tuesday, March 2, 2021 3:29 PM

$K \subseteq \mathbb{R}^n$ convex. $f : K \to \mathbb{R}$ convex.

$$\min_{x \in K} f(x)$$

Projected gradient descent: Start with $x_0 \in K$.
For T steps, $x_{t+1} := \Pi_K(x_t - \eta \nabla f(x_t))$

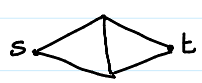
Assumptions: ① f L-smooth on K.
$\forall\, x, y \in K$, $\|\nabla f(x) - \nabla f(y)\|_2 \le L \|x - y\|_2$

② $\|x_0 - x^*\|_2 \le D$

Theorem: Projected gradient descent with $T = O\left(\frac{LD^2}{\varepsilon}\right)$ iterations outputs $x_T \in K$ s.t. $f(x_T) - f(x^*) \le \varepsilon$.

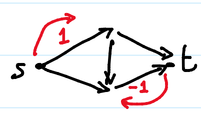
Application: Max flow

Undirected graph: $G = (V, E)$, source s, sink t, $s \neq t \in V$.



Each edge has a unit capacity

Goal: Send the maximum amount of flow from s to t while obeying the edge constraints.



Consider an arbitrary orientation of G.
A flow is a vector $x \in \mathbb{R}^E$ s.t.

$$\underbrace{\sum_{e : e = (u,v)} x_e}_{\text{inflow}} = \underbrace{\sum_{e : e = (v,w)} x_e}_{\text{outflow}} \quad \forall\, v \in V \setminus \{s, t\}.$$

value of the flow $F = \sum_{e : e = (s,v)} x_e - \sum_{e : e = (u,s)} x_e$

Linear program:

$$\max_{x \in \mathbb{R}^E, F} F$$

$$\text{s.t. } Bx = F(e_s - e_t)$$

$$|x_i| \le 1 \ \forall\, i \in [m].$$

$B \in \mathbb{R}^{n \times m}$ vertex-edge incidence matrix

$B_{u,(u,v)} = 1$

$B_{v,(u,v)} = -1$

all other entries zero

Simpler question: is there a flow of value F?

<u>simpler question:</u> is there a flow of value F ?

Find $x \in \mathbb{R}^m$

s.t. $Bx = F(e_s - e_T)$

$\|x\|_\infty \le 1$.

$H_F := \{ x \in \mathbb{R}^m : Bx = F(e_s - e_T) \}$ $\qquad K := H_F \cap B_{m,\infty}$

$B_{m,\infty} := \{ x \in \mathbb{R}^m : \|x\|_\infty \le 1 \}$

<u>Two convex opt formulations:</u>

① Minimize the distance to $B_{m,\infty}$ while constraining the point to be in H_F .

② The other option

$P : \mathbb{R}^m \to B_{m,\infty}$

$P(x) := \operatorname{argmin} \{ \|x - y\|_2 : y \in B_{m,\infty} \}$.

$$\to \min_{x \in \mathbb{R}^E} \|x - P(x)\|_2^2$$

$$\text{s.t. } Bx = F(e_s - e_T)$$

① <u>Convexity:</u> convex set $S \subseteq \mathbb{R}^n$, $x \to \operatorname{dist}(x, S)^2$ is convex.

$\operatorname{dist}(x, S) := \inf_{y \in S} \|x - y\|_2$ $\qquad$ exercise

② <u>Smoothness:</u> $P(x)_i = \begin{cases} -1 , & \text{if } x_i < -1 \\ x_i & \text{if } x_i \in [-1, 1] \\ 1 & \text{if } x_i > 1 \end{cases}$

$\nabla f(x)_i = \begin{cases} 2(x_i + 1) & \text{if } x_i < -1 \\ 0 & \text{if } x_i \in [-1, 1] \\ 2(x_i - 1) & \text{if } x_i > 1 \end{cases}$

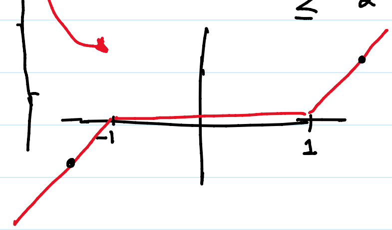
$f(x) := \|x - P(x)\|_2^2 = \sum_{i=1}^{n} h(x)_i$

$h(z) = \begin{cases} (z+1)^2 & \text{if } z < -1 \\ 0 & \text{if } z \in [-1, 1] \\ (z-1)^2 & \text{if } z > 1 \end{cases}$

<u>Claim:</u> $|\nabla f(x)_i - \nabla f(y)_i| \le 2 |x_i - y_i|$

$\|\nabla f(x) - \nabla f(y)\|_2 = \sqrt{\sum_i (\nabla f(x)_i - \nabla f(y)_i)^2}$

$\overset{\text{claim}}{\le} 2 \|x - y\|_2$



③ <u>Good starting point:</u> want to find x s.t. $Bx = F(e_s - e_T)$

& x is as close to $B_{m,\infty}$ as possible

(it has as small $\|x\|_\infty$ as possible).

$\operatorname{argmin} \|x - P(x)\|_2$

$-(e_s - e_T)$

& x is as close to $1/m, \infty$ as possible
(it has as small $\|x\|_\infty$ as possible).

$$x^* = \arg\min \|x - p(x)\|_2 \quad \text{s.t. } Bx = F(e_S - e_T)$$

find $\min_x \|x\|_2$
s.t. $Bx = F(e_S - e_T)$. → this can be done efficiently ($\tilde{O}(m)$) using Laplacian linear system solvers.

g is the solution

→ if there is a flow with value F & $\|x\|_\infty \le 1$. $\Rightarrow \|g\|_2^2 \le m$.

$\Rightarrow \|g\|_2 \le \sqrt{m}$

For x^* also, $\|x^*\|_2 \le \sqrt{m}$

$$\Rightarrow \|g - x^*\|_2 \le 2\sqrt{m} \quad D$$

(4) Gradient computation & projection oracle : → Gradient computation cheap
→ Projection → $\tilde{O}(m)$ using Laplacian solver.

$L = 2$, $D = 2\sqrt{m}$, cost of each iteration $= \tilde{O}(m)$.
\# iterations : $O\left(\frac{LD^2}{\delta}\right)$ to find an δ-approx. sol^n.

Run time : $\tilde{O}\left(\frac{m^2}{\delta}\right)$ to find δ-approx sol^n to
$$\min \|x - p(x)\|_2^2 \quad \text{s.t. } Bx = F(e_S - e_T)$$

Thm : There is an algorithm which given an undirected graph $G = (V, E)$, two special vertices s, t, unit capacities, finds a flow of value $\ge (1-\varepsilon)F^*$ in time $\tilde{O}\left(\frac{m^{2.5}}{\varepsilon F^*}\right)$.

Is gradient descent optimal ?

$f: \mathbb{R}^n \to \mathbb{R}$ convex. f is L-smooth, $\|x^*\|_2 \le D$.

First order oracle : $x \to$ oracle $\to f(x), \nabla f(x)$

Question: How many oracle calls do we require to output some x s.t. $f(x)-f(x^*) \le \varepsilon$?

Gradient descent: requires $O\left(\frac{LD^2}{\varepsilon}\right)$ calls.

Turns out the right answer is $\Theta\left(\sqrt{\frac{LD^2}{\varepsilon}}\right)$.

Thm [lower bound]: There is a family of functions $\{f\}$ with smoothness L, diameter bound D s.t. any randomized alg will require $\Omega\left(\sqrt{\frac{LD^2}{\varepsilon}}\right)$ oracle calls to output an ε-suboptimal point.

Nesterov, Nemirovski, ...

Thm [upper bound]: There is an algorithm which finds an ε-suboptimal point in $O\left(\sqrt{\frac{LD^2}{\varepsilon}}\right)$ iterations.

Nesterov's accelerated gradient descent.

Lower bound also extends to quantum algorithms [GKNS 21].