Convex Optimization

Prof. Nati Srebro

Lecture 16:

Lower Bounds

Bubeck Section 3.5

Stochastic Approximation

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ s.t. \quad & x \in K \end{aligned}$$

Method	Oracle	Assumptions	# accesses	Adtl. runtime
Center of Mass	1st + Separation	$\lvert f \rvert \leq B$	$O\left(n \log \frac{B}{\epsilon}\right)$	NA
Ellipsoid			$O\left(n^2 \log \frac{B}{\epsilon}\right)$	$O\left(n^4 \log \frac{B}{\epsilon}\right)$
Vaidya++			$\tilde{O}\left(n \log \frac{B}{\epsilon}\right)$	$\tilde{O}\left(n^3 \log \frac{B}{\epsilon}\right)$

$$\begin{aligned} \min_{x \in \mathbb{R}} \quad & f(x) \\ s.t. \quad & -R \leq x \leq R \end{aligned}$$

For any algorithm A that uses a 1st order oracle and any ϵ, there exists a convex $f: [-1,1] \to \mathbb{R}$, $|f(x)| \leq 1$, such that on input f, A calls the oracle at least $\frac{1}{5}\log_2 \frac{1}{\epsilon} - 1$ times before returning an ϵ-suboptimal point.

By scaling $\tilde{f}(x) = B \cdot f(x/R)$:

For any algorithm A that uses a 1st order oracle and any B,R,ϵ, there exists a convex $f: [-R, R] \to \mathbb{R}$, $|f(x)| \leq B$, such that on input f, A calls the oracle at least $\frac{1}{5}\log_2 \frac{B}{\epsilon} - 1$ times before returning an ϵ-suboptimal point.

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ s.t. \quad & \|x\| \leq R \end{aligned}$$

Theorem (requires more complex adversary constructing piecewise linear $f: \mathbb{R}^n \to \mathbb{R}$): for any algorithm A that uses a 1st order oracle and any B, R, ϵ, there exists a convex $f: \mathbb{R}^n \to \mathbb{R}$, $|f(x)| \leq B$, such that on input f, A calls the oracle at least $\Omega\left(n \log \frac{B}{\epsilon}\right)$ times before returning an ϵ-suboptimal point.

Lower bound holds for 2nd, even 3rd, or any "local" oracle.

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f_0(x) \\ s.t. \quad & f_i(x) \leq 0 \end{aligned}$$

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f_0(x) \\ s.t. \quad & x \in K \end{aligned}$$

Problem is more general, so lower bound holds.

Enough to show $\forall_A \exists_{(f_0, f_i)} T\big(A, (f_0, f_i)\big) > T$ or $\forall_A \exists_{(f_0, K)} T\big(A, (f_0, K)\big) > T$

Take $f_i(x) = 1 - \|x\|^2$ or $K = \{x | \|x\| \leq R\}$, and "hard" f_0 as above.

Method	Oracle	Assumptions	# accesses	Adtl. runtime
Center of Mass	1st	$\lvert f \rvert \leq B$	$O\left(n \log \frac{B}{\epsilon}\right)$	NA
Ellipsoid			$O\left(n^2 \log \frac{B}{\epsilon}\right)$	$O\left(n^4 \log \frac{B}{\epsilon}\right)$
Vaidya++			$\tilde{O}\left(n \log \frac{B}{\epsilon}\right)$	$\tilde{O}\left(n^3 \log \frac{B}{\epsilon}\right)$
Grad Descent		$\lVert \nabla f \rVert \leq L$ $\lVert x^* \rVert \leq R$	$O\left(\frac{L^2 R^2}{\epsilon^2}\right)$	$O\left(n \frac{L^2 R^2}{\epsilon^2}\right)$

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ s.t. \quad & \|x\|_2 \le 1 \end{aligned}$$

Assumption: $\|\nabla f(x)\|_2 \le L$

- Consider very high dim, $n > 2\frac{L^2}{\epsilon^2}$ (otherwise can use center-of-mass)
- Construct a "hard" function of the form $f(x) = L \cdot \max_i \langle v_i, x\rangle$

  ➔ $\nabla f(x) = Lv_i$ s.t. $j = \arg\max_i \langle v_i, x\rangle$
- Given algorithm A, choose v_i adversarily by simulating A:

  On query $x^{(k)}$, pick $v_k \perp x^{(1)}, \dots, x^{(k)}, v_1, \dots, v_{k-1}$, $\|v_k\| = 1$

  answer using $f^{(k)}(x) = L \cdot \max_{i=1..k} \langle v_i, x\rangle$

Claim: $f^{(k)}(x)$ are convex and L-Lipschitz

Claim: $f^{(k)}\left(x^{(k)}\right) \ge 0$

Claim: For all $i \le k$, answer on $x^{(i)}$ using $f^{(i)}$ also valid for $f^{(k)}$

Proof: For $i < j \le k$, $\left\langle v_j, x^{(i)}\right\rangle = 0 \le f^{(i)}\left(x^{(i)}\right)$ so can ignore in max and argmax

Consider $x^* = \frac{-1}{\sqrt{T}}\sum_{i=1}^{T} v_i$, then $\|x^*\| = 1$ and $f^{(T)}(x^*) = \frac{-L}{\sqrt{T}}$

➔ $\min_{i=1..T} f^{(T)}\left(x^{(i)}\right) \ge 0 \ge f^{(T)}(x^*) + \frac{L}{\sqrt{T}}$

Conclusion: A requires $\ge L^2/\epsilon^2$ queries to find ϵ-suboptimal of f

Method	Oracle	Assumptions	# accesses	Adtl. runtime
Center of Mass	1st	$\lvert f \rvert \le B$	$O\left(n \log \frac{B}{\epsilon}\right)$	NA
Ellipsoid			$O\left(n^2 \log \frac{B}{\epsilon}\right)$	$O\left(n^4 \log \frac{B}{\epsilon}\right)$
Vaidya++			$\tilde{O}\left(n \log \frac{B}{\epsilon}\right)$	$\tilde{O}\left(n^3 \log \frac{B}{\epsilon}\right)$
Grad Descent		$\mu \preccurlyeq \nabla^2 f \preccurlyeq M$ $\kappa = M/\mu$ $\lvert f \rvert \le B$	$O(\kappa \log B/\epsilon)$	$O(n\kappa \log B/\epsilon)$
Accelerated GD			$O(\sqrt{\kappa} \log B/\epsilon)$	$O(n\sqrt{\kappa} \log B/\epsilon)$
Grad Descent		$\nabla^2 f \preccurlyeq M$ $\lVert x^* \rVert \le R$	$O\left(\frac{MR^2}{\epsilon}\right)$	$O\left(n\frac{MR^2}{\epsilon}\right)$
Accelerated GD			$O\left(\sqrt{\frac{MR^2}{\epsilon}}\right)$	$O\left(n\sqrt{\frac{MR^2}{\epsilon}}\right)$
Grad Descent		$\lVert \nabla f \rVert \le L$ $\lVert x^* \rVert \le R$	$O\left(\frac{L^2R^2}{\epsilon^2}\right)$	$O\left(n\frac{L^2R^2}{\epsilon^2}\right)$
			can't improve	

Oracle Lower Bounds

- Unconditional and exact (not just big-O or “poly”)
- Bound number of oracle accesses, not computation
- Can’t tell us whether there is a *computational efficient* method
- For specific problem (eg LP), maybe better based on direct access

Method	Oracle	Assumptions	# accesses	Adtl. runtime
Center of Mass	1st	$\lvert f \rvert \le B$	$O\left(n \log \frac{B}{\epsilon}\right)$	NA
Ellipsoid			$O\left(n^2 \log \frac{B}{\epsilon}\right)$	$O\left(n^4 \log \frac{B}{\epsilon}\right)$
Vaidya++			$\tilde{O}\left(n \log \frac{B}{\epsilon}\right)$	$\tilde{O}\left(n^3 \log \frac{B}{\epsilon}\right)$
Grad Descent		$\mu \preccurlyeq \nabla^2 f \preccurlyeq M$ $\kappa = M/\mu$ $\lvert f \rvert \le B$	$O(\kappa \log B/\epsilon)$	$O(n\kappa \log B/\epsilon)$
Accelerated GD			$O(\sqrt{\kappa} \log B/\epsilon)$	$O(n\sqrt{\kappa} \log B/\epsilon)$
Grad Descent		$\nabla^2 f \preccurlyeq M$ $\lVert x^* \rVert \le R$	$O\left(\frac{MR^2}{\epsilon}\right)$	$O\left(n\frac{MR^2}{\epsilon}\right)$
Accelerated GD			$O\left(\sqrt{\frac{MR^2}{\epsilon}}\right)$	$O\left(n\sqrt{\frac{MR^2}{\epsilon}}\right)$
Grad Descent		$\lVert \nabla f \rVert \le L$ $\lVert x^* \rVert \le R$	$O\left(\frac{L^2R^2}{\epsilon^2}\right)$	$O\left(n\frac{L^2R^2}{\epsilon^2}\right)$
			can't improve	

Faster than Gradient Descent?

$$\min_{x\in\mathbb{R}^n} \quad F(x) = \frac{1}{m}\sum_{i=1}^{m} f_i(x)$$

- E.g. $f_i(x) = |\langle a_i, x\rangle - b_i|$, $F(x) = \frac{1}{m}\|Ax - b\|_1$, assume $\|a_i\| \le 1$

$$\nabla F(x) = \frac{1}{m}\sum_{i=1}^{m} \nabla f_i(x) = \frac{1}{m}\sum_{i=1}^{m} sign(\langle a_i, x\rangle - b_i)a_i$$

$$x^{(k+1)} \leftarrow x^{(k)} - \frac{t^{(k)}}{m}\sum_{i=1}^{m} sign\left(\left\langle a_i, x^{(k)}\right\rangle - b_i\right)a_i$$

- Runtime:

$$\text{(\#iter)} \cdot \text{(runtime per iteration)} = O\left(\frac{R^2}{\epsilon^2} nm\right)$$

$L = \sup_x \|\nabla F(x)\| \le 1$, $\|x^*\| \le R$

$$\frac{L^2R^2}{\epsilon^2} \qquad nm$$

Stochastic Gradient Descent

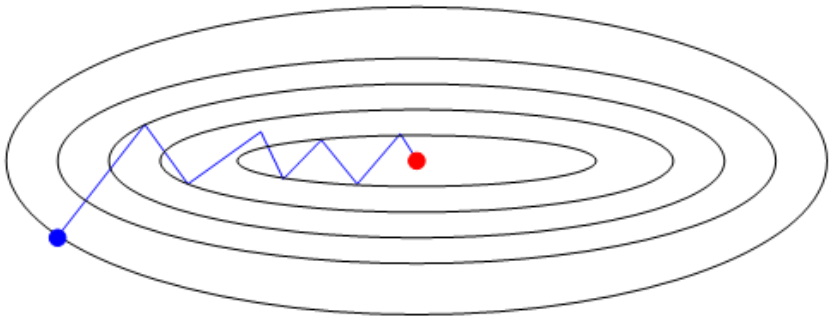
$$\min_{x \in \mathbb{R}^n} \quad F(x) = \frac{1}{m} \sum_{i=1}^{m} f_i(x)$$

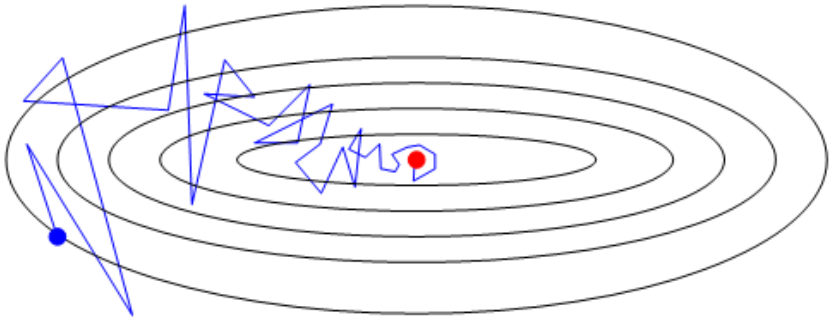
Instead of $\nabla F\left(x^{(k)}\right)$ enough to use unbiased estimator

$$g^{(k)} \quad \text{s.t.} \quad \mathbb{E}\left[g^{(k)}\right] = \nabla F\left(x^{(k)}\right) \qquad \left(\text{or } \mathbb{E}\left[g^{(k)}\right] \in \partial F\left(x^{(k)}\right)\right)$$

Init	$x^{(0)} = 0$
Iterate	obtain $g^{(k)}$ s.t. $\mathbb{E}\left[g^{(k)}\right] \in \partial F\left(x^{(k)}\right)$
	$x^{(k+1)} \leftarrow x^{(k)} - t^{(k)} g^{(k)}$
Return	$\tilde{x}^{(k)} = \frac{1}{k} \sum_{i=1}^{k} x^{(i)}$

GD **SGD**





Stochastic Gradient Descent

$$\min_{x \in \mathbb{R}^n} \quad F(x) = \frac{1}{m} \sum_{i=1}^{m} f_i(x)$$

Instead of $\nabla F\left(x^{(k)}\right)$ enough to use unbiased estimator

$$g^{(k)} \quad \text{s.t.} \quad \mathbb{E}\left[g^{(k)}\right] = \nabla F\left(x^{(k)}\right) \qquad \left(\text{or } \mathbb{E}\left[g^{(k)}\right] \in \partial F\left(x^{(k)}\right)\right)$$

Init	$x^{(0)} = 0$
Iterate	obtain $g^{(k)}$ s.t. $\mathbb{E}\left[g^{(k)}\right] \in \partial F\left(x^{(k)}\right)$
	$x^{(k+1)} \leftarrow x^{(k)} - t^{(k)} g^{(k)}$
Return	$\tilde{x}^{(k)} = \frac{1}{k} \sum_{i=1}^{k} x^{(i)}$

Required oracle: **stochastic** 1st order oracle, $x \mapsto g$ s.t. $\mathbb{E}[g|x] \in \partial F(x)$ and each invocation is independent.

Theorem: If F is convex, $\forall_x \mathbb{E}[\|g\|^2|x] \leq L^2$ and $\|x^*\| \leq R$, then

using $t^{(k)} = \frac{R}{L\sqrt{k+1}}$: $\mathbb{E}\left[F\left(\tilde{x}^{(k)}\right)\right] \leq p^* + \sqrt{\frac{R^2 L^2}{k}}$ (i.e. need $k = O\left(\frac{R^2 L^2}{\epsilon^2}\right)$ iter)

Stochastic Gradient Descent

$$\min_{x \in \mathbb{R}^n} \quad F(x) = \frac{1}{m} \sum_{i=1}^{m} f_i(x)$$

Instead of $\nabla F\left(x^{(k)}\right)$ enough to use unbiased estimator

$$g^{(k)} \quad \text{s.t.} \quad \mathbb{E}\left[g^{(k)}\right] = \nabla F\left(x^{(k)}\right) \qquad \left(\text{or } \mathbb{E}\left[g^{(k)}\right] \in \partial F\left(x^{(k)}\right)\right)$$

How can we get a stochastic 1st order oracle in the example above?

$$x \mapsto \nabla F(x) \quad \text{or} \quad x \mapsto \nabla F(x) + \mathcal{N}(0, \sigma^2)$$

Better: $x \mapsto g = \nabla f_i(x)$ for $i \sim \text{Unif}[1..m]$

$$\mathbb{E}[g] = \sum_i P(i) \nabla f_i(x) = \sum_i \frac{1}{m} \nabla f_i(x) = \nabla F(x)$$

Init	$x^{(0)} = 0$
Iterate	$i \sim \text{Unif}[1..m]$
	$x^{(k+1)} \leftarrow x^{(k)} - t^{(k)} \nabla f_i\left(x^{(k)}\right)$
Return	$\tilde{x}^{(k)} = \frac{1}{k} \sum_{i=1}^{k} x^{(i)}$

$$\min_{x \in \mathbb{R}^n} \quad \frac{1}{m} \sum_{i=1}^{m} |\langle a_i, x \rangle - b_i|$$

GD:

$$x^{(k+1)} \leftarrow x^{(k)} - \frac{t^{(k)}}{m} \sum_{i=1}^{m} sign\left(\langle a_i, x^{(k)} \rangle - b_i\right) a_i$$

Runtime: (#iter) · (runtime per iteration) $= O\left(\frac{R^2}{\epsilon^2} nm\right)$

SGD:

$$i \sim \text{Unif}[1..m]$$
$$x^{(k+1)} \leftarrow x^{(k)} - t^{(k)} sign\left(\langle a_i, x^{(k)} \rangle - b_i\right) a_i$$

$$g = \nabla f_i(x) \text{ for } i \sim \text{Unif}[1..m]$$

$\frac{L^2 R^2}{\epsilon^2}$

$L = \sup_x \mathbb{E}[\|g\|^2 | x] = \mathbb{E}_i[\|a_i\|^2] \leq 1$

$\|x^*\| \leq R$

Runtime: (#iter) · (runtime per iteration) $= O\left(\frac{R^2}{\epsilon^2} n\right)$

$O(n)$

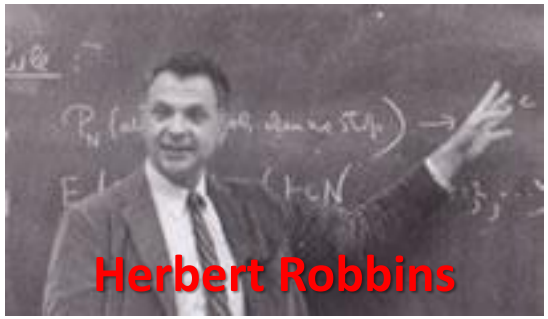
Stochastic Gradient Descent

$$\min_{x \in \mathbb{R}^n} \quad F(x)$$

Init	$x^{(0)} = 0$
Iterate	obtain $g^{(k)}$ s.t. $\mathbb{E}\left[g^{(k)}\right] \in \partial F\left(x^{(k)}\right)$
	$x^{(k+1)} \leftarrow x^{(k)} - t^{(k)} g^{(k)}$
Return	$\tilde{x}^{(k)} = \frac{1}{k}\sum_{i=1}^{k} x^{(i)}$

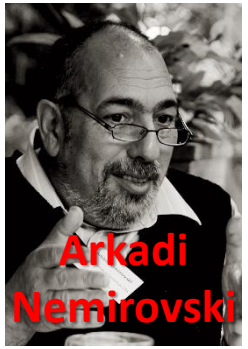
Required oracle: **stochastic** 1st order oracle, $x \mapsto g$ s.t. $\mathbb{E}[g|x] \in \partial F(x)$ and each invocation is independent.

$\mathbb{E}[\|g\|^2] \leq L^2$ and $\|x^*\| \leq R$ ➔ $\mathbb{E}\left[F\left(\tilde{x}^{(k)}\right)\right] \leq p^* + \sqrt{\frac{R^2 L^2}{k}}$



1952 as “Stochastic Approximation”

Sutton Monro



1978

Stochastic Gradient Descent

$$\min_{x \in \mathbb{R}^n} \quad F(x)$$

Init	$x^{(0)} = 0$
Iterate	obtain $g^{(k)}$ s.t. $\mathbb{E}\left[g^{(k)}\right] \in \partial F\left(x^{(k)}\right)$
	$x^{(k+1)} \leftarrow x^{(k)} - t^{(k)} g^{(k)}$
Return	$\tilde{x}^{(k)} = \frac{1}{k}\sum_{i=1}^{k} x^{(i)}$

Required oracle: **stochastic** 1st order oracle, $x \mapsto g$ s.t. $\mathbb{E}[g|x] \in \partial F(x)$ and each invocation is independent.

$\mathbb{E}[\|g\|^2] \leq L^2$ and $\|x^*\| \leq R$ ➔ $\mathbb{E}\left[F\left(\tilde{x}^{(k)}\right)\right] \leq p^* + \sqrt{\frac{R^2 L^2}{k}}$

$\mathbb{E}[\|g\|^2] \leq L^2$ and $\mu \preccurlyeq \nabla^2 F$ ➔ $\mathbb{E}\left[F\left(\tilde{x}^{(k)}\right)\right] \leq p^* + \frac{L}{\mu k}$

$\mathbb{E}[\|g - \nabla f(x)\|^2] \leq \sigma^2$, $\|x^*\| \leq R$ and $\nabla^2 F \preccurlyeq M$

➔ $\mathbb{E}\left[F\left(\tilde{x}^{(k)}\right)\right] \leq p^* + \frac{MR^2}{k} + \sqrt{\frac{R^2\sigma^2}{k}}$

Stochastic Gradient Descent

$$\min_{x \in \mathbb{R}^n} \quad F(x) = \frac{1}{m} \sum_{i=1}^{m} f_i(x)$$

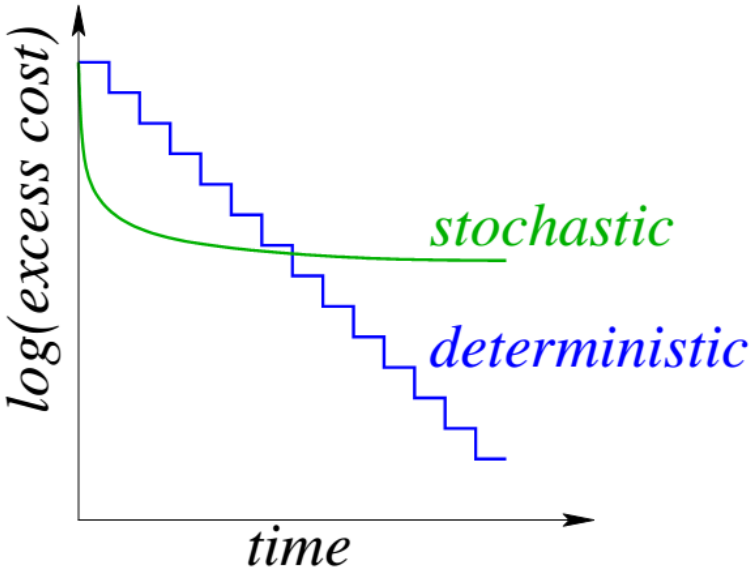
$$\mu \preccurlyeq \nabla^2 f_i(x) \preccurlyeq M$$

Gradient descent:

(#iter) · (runtime per iteration) $= O\left(\frac{M}{\mu} \log \frac{F(x^{(0)}) - p^*}{\epsilon} \cdot \boldsymbol{m} \cdot \nabla f_i \text{ computes}\right)$

Stochastic gradient descent:

(#iter) · (runtime per iteration) $= O\left(\frac{\|\nabla f_i\|}{\mu} \frac{1}{\epsilon} \cdot \mathbf{1} \cdot \nabla f_i \text{ computes}\right)$



Stochastic Gradient Descent

$$\min_{x \in \mathbb{R}^n} \quad F(x) = \frac{1}{m} \sum_{i=1}^{m} f_i(x)$$

$$\mu \preccurlyeq \nabla^2 f_i(x) \preccurlyeq M$$

Gradient descent:

(#iter) · (runtime per iteration) $= O\left(\frac{M}{\mu} \log \frac{F(x^{(0)}) - p^*}{\epsilon} \cdot \boldsymbol{m} \cdot \nabla f_i \text{ computes}\right)$

Stochastic gradient descent:

(#iter) · (runtime per iteration) $= O\left(\frac{\|\nabla f_i\|}{\mu} \frac{\mathbf{1}}{\epsilon} \cdot \mathbf{1} \cdot \nabla f_i \text{ computes}\right)$

SDCA [Shalev-Shwartz Zhang 2013], SAG [Schmidt Le-Roux Bach 2013], SVRG [Johnson Zhang 2013]:

$$O\left(\left(\frac{M}{\mu} + m\right) \log \frac{1}{\epsilon}\right)$$

With acceleration: $O\left(\left(\sqrt{\frac{M}{\mu} m} + m\right) \log \frac{1}{\epsilon}\right)$

[Blake 2016]: This is the best* possible! (matching lower bound)

*up to a log-factor

Stochastic Coordinate Descent

$$\min_{x \in \mathbb{R}^n} \quad F(x)$$

Init	$x^{(0)} = 0$
Iterate	Pick $i \sim \mathrm{Unif}[1..n]$
	$x^{(k+1)} \leftarrow x^{(k)} - t^{(k)} \left(\frac{\partial}{\partial x_i} F\left(x^{(k)}\right) \right) e_i$

Stochastic gradient descent with stochastic 1st order oracle:

$$x \mapsto n \left(\frac{\partial}{\partial x_i} F(x) \right) e_i \quad i \sim \mathrm{Unif}[1..n]$$

$$\min_{x \in \mathbb{R}^n} \quad \frac{1}{m} \sum_{i=1}^{m} |\langle a_i, x \rangle - b_i| + \frac{\lambda}{2} \|x\|^2$$

- Option 1: rewrite as QP, use IP methods:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \frac{1}{m} \sum_{i=1}^{m} z_i + \frac{\lambda}{2} \|x\|^2 \\ \text{s.t.} \quad & -z \leq \langle a_i, x \rangle - b_i \leq z \end{aligned}$$

Runtime: $O\left(\sqrt{m} n^3 \log \frac{1}{\epsilon}\right)$

- Option 2: (sub)gradient descent on non-smooth objective

$$x^{(k+1)} \leftarrow x^{(k)} - \frac{1}{\lambda k m} \sum_{i=1}^{m} sign(\langle a_i, x \rangle - b_i) a_i - \frac{1}{k} x$$

Runtime: $O\left(\frac{1}{\lambda \epsilon} nm\right)$

- Option 3: Stochasitc (sub)gradient descent on non-smooth objective

Pick $i \sim \text{Unif}[1..n]$

$$x^{(k+1)} \leftarrow x^{(k)} - \frac{1}{\lambda k} sign(\langle a_i, x \rangle - b_i) a_i - \frac{1}{k} x$$

Runtime: $O\left(\frac{1}{\lambda \epsilon} n\right)$