Limits on gradient compression for stochastic optimization

Prathamesh Mayekar

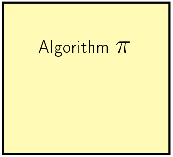
Joint work with

Himanshu Tyagi

Department of ECE,
Indian Institute of Science

The Setup

Classical Setup [1]

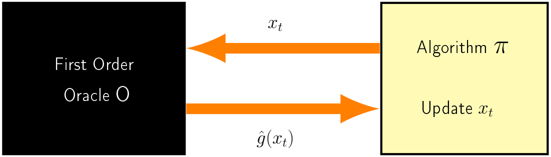


Algorithm π:

- Input: Domain $\mathcal{X}$, function and oracle class $\mathcal{O}$
- Goal: Minimize unknown function f using an oracle O, where $\{f, O\}$ belong to $\mathcal{O}$.

[1] Nemirovsky, A. S., and Yudin, D. B. (1983). Problem complexity and method efficiency in optimization.

Classical Setup [1]



Algorithm π:

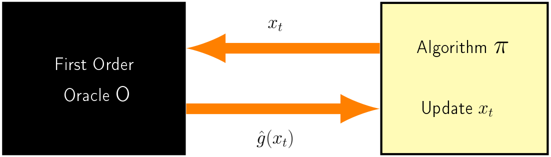
- Input: Domain $\mathcal{X}$, function and oracle class $\mathcal{O}$
- Goal: Minimize unknown function f using an oracle O, where $\{f, O\}$ belong to $\mathcal{O}$.

First Order Oracle O:

- Returns a noisy sub-gradient estimate $\hat{g}(x_t)$ for query x_t.

[1] Nemirovsky, A. S., and Yudin, D. B. (1983). Problem complexity and method efficiency in optimization.

Classical Setup [1]



Algorithm π:

- Input: Domain $\mathcal{X}$, function and oracle class $\mathcal{O}$
- Goal: Minimize unknown function f using an oracle O, where $\{f, O\}$ belong to $\mathcal{O}$.

First Order Oracle O:

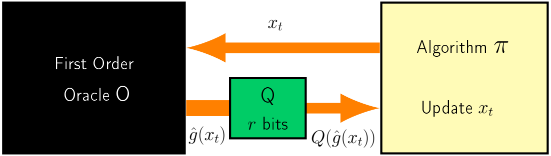
- Returns a noisy sub-gradient estimate $\hat{g}(x_t)$ for query x_t.

Main Question:

Which π gives the best convergence rate?

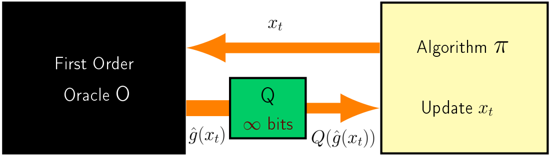
[1] Nemirovsky, A. S., and Yudin, D. B. (1983). Problem complexity and method efficiency in optimization.

Our Refinement



$\hat{g}(x_t)$ can be sent to a finite precision r using a Q of our choice.

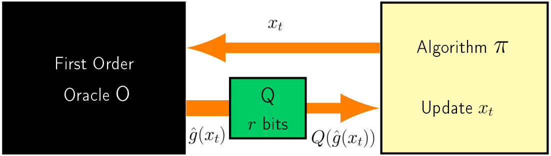
Our Refinement



$\hat{g}(x_t)$ can be sent to a finite precision r using a Q of our choice.

Reduces to classical setup if we are allowed infinite precision.

Our Refinement



$\hat{g}(x_t)$ can be sent to a finite precision r using a Q of our choice.

Reduces to classical setup if we are allowed infinite precision.

Main Question:

What is the minimum r to attain the convergence rate of classic case?

ℓ_p optimization family

ℓ_p optimization family

Assumptions:

- Domain $\mathcal{X}$ will be the ℓ_p ball of diameter D in $\mathbb{R}^d$.

ℓ_p optimization family

Assumptions:

- Domain $\mathcal{X}$ will be the ℓ_p ball of diameter D in $\mathbb{R}^d$.
- Function, Oracle class $\mathcal{O}_p$ consists of all tuples $\{f, O\}$ such that
  1. f is convex.

ℓ_p optimization family

Assumptions:

- Domain $\mathcal{X}$ will be the ℓ_p ball of diameter D in $\mathbb{R}^d$.
- Function, Oracle class $\mathcal{O}_p$ consists of all tuples $\{f, O\}$ such that
  1. f is convex.
  2. *Unbiased*: $\mathbb{E}\left[\hat{g}(x)|x\right] \in \partial f(x)$.

ℓ_p optimization family

Assumptions:

- Domain $\mathcal{X}$ will be the ℓ_p ball of diameter D in $\mathbb{R}^d$.
- Function, Oracle class $\mathcal{O}_p$ consists of all tuples $\{f, O\}$ such that
  1. f is convex.
  2. *Unbiased*: $\mathbb{E}\left[\hat{g}(x)|x\right] \in \partial f(x)$.
  3. *Almost surely norm-bounded*: $\|\hat{g}(x)\|_q \leq B$, where $q = \frac{p}{p-1}$.

ℓ_p optimization family

Assumptions:

- Domain $\mathcal{X}$ will be the ℓ_p ball of diameter D in $\mathbb{R}^d$.
- Function, Oracle class $\mathcal{O}_p$ consists of all tuples $\{f, O\}$ such that
  1. f is convex.
  2. *Unbiased*: $\mathbb{E}\left[\hat{g}(x)|x\right] \in \partial f(x)$.
  3. *Almost surely norm-bounded*: $\left\|\hat{g}(x)\right\|_q \leq B$, where $q = \dfrac{p}{p-1}$.

- Minmax optimization accuracy

$$\mathcal{E}(T, r, p) := \inf_{\pi \in \Pi_T} \inf_{Q \in \mathcal{Q}_r} \sup_{\{f, O\} \in \mathcal{O}} \mathbb{E}\left[f(x(\pi, Q))\right] - f^*.$$

ℓ_p optimization family

Assumptions:

- Domain $\mathcal{X}$ will be the ℓ_p ball of diameter D in $\mathbb{R}^d$.
- Function, Oracle class $\mathcal{O}_p$ consists of all tuples $\{f, O\}$ such that
  1. f is convex.
  2. *Unbiased*: $\mathbb{E}\left[\hat{g}(x)|x\right] \in \partial f(x)$.
  3. *Almost surely norm-bounded*: $\|\hat{g}(x)\|_q \leq B$, where $q = \dfrac{p}{p-1}$.
- Minmax optimization accuracy

$$\mathcal{E}(T, r, p) := \inf_{\pi \in \Pi_T} \inf_{Q \in \mathcal{Q}_r} \sup_{\{f,O\} \in \mathcal{O}} \mathbb{E}\left[f(x(\pi, Q))\right] - f^*.$$

- Classical Result: $\mathcal{E}(T, \infty, p) = \tilde{\Theta}\left(\dfrac{(d^{1/2-1/p} \wedge 1)DB}{\sqrt{T}}\right)$.

ℓ_p optimization family

Assumptions:

- Domain $\mathcal{X}$ will be the ℓ_p ball of diameter D in $\mathbb{R}^d$.
- Function, Oracle class $\mathcal{O}_p$ consists of all tuples $\{f, O\}$ such that
  1. f is convex.
  2. *Unbiased*: $\mathbb{E}\left[\hat{g}(x)|x\right] \in \partial f(x)$.
  3. *Almost surely norm-bounded*: $\|\hat{g}(x)\|_q \leq B$, where $q = \dfrac{p}{p-1}$.
- Minmax optimization accuracy

$$\mathcal{E}(T, r, p) := \inf_{\pi \in \Pi_T} \inf_{Q \in \mathcal{Q}_r} \sup_{\{f, O\} \in \mathcal{O}} \mathbb{E}\left[f(x(\pi, Q))\right] - f^*.$$

- We will characterize

$$r^*(T, p) := \min\{r : \mathcal{E}(T, r, p) \approx \mathcal{E}(T, \infty, p)\},$$

minimum precision at which the composed oracle starts behaving like the classic, unresticted oracle.

Characterizing $r^*(T, p)$

Lower Bound

Theorem

1. For $1 \leq p < 2$,

$$r^*(T,p) \gtrapprox d.$$

Lower Bound

Theorem

1. For $1 \le p < 2$,
$$r^*(T,p) \gtrapprox d.$$
2. For $2 \le p$,
$$r^*(T,p) \gtrapprox d^{\frac{2}{p}} \vee \log d.$$

Lower Bound

Theorem

1. For $1 \leq p < 2$,
$$r^*(T,p) \gtrsim d.$$
2. For $2 \leq p$,
$$r^*(T,p) \gtrsim d^{\frac{2}{p}} \vee \log d.$$

- Techniques from [Agarwal et al. 12], [Mayekar et al. 20].

Lower Bound

Theorem

1. For $1 \leq p < 2$,
$$r^*(T,p) \gtrapprox d.$$

2. For $2 \leq p$,
$$r^*(T,p) \gtrapprox d^{\frac{2}{p}} \vee \log d.$$

- Techniques from [Agarwal et al. 12], [Mayekar et al. 20].
- We construct "difficult" oracles for optimization, compression.

Lower Bound

Theorem

1. For $1 \leq p < 2$,
$$r^*(T,p) \gtrapprox d.$$

2. For $2 \leq p$,
$$r^*(T,p) \gtrapprox d^{\frac{2}{p}} \vee \log d.$$

- Techniques from [Agarwal et al. 12], [Mayekar et al. 20].
- We construct "difficult" oracles for optimization, compression.
- For $p \in [1,2)$, the same oracle is "difficult" for optimization, compression.

Lower Bound

Theorem

1. For $1 \leq p < 2$,

$$r^*(T, p) \gtrsim d.$$

2. For $2 \leq p$,

$$r^*(T, p) \gtrsim d^{\frac{2}{p}} \vee \log d.$$

- Techniques from [Agarwal et al. 12], [Mayekar et al. 20].
- We construct “difficult” oracles for optimization, compression.
- For $p \in [1, 2)$, the same oracle is “difficult” for optimization, compression.
- For $p \geq 2$, these two oracles differ:

Lower Bound

Theorem

1. For $1 \leq p < 2$,
$$r^*(T, p) \gtrsim d.$$
2. For $2 \leq p$,
$$r^*(T, p) \gtrsim d^{\frac{2}{p}} \vee \log d.$$

- Techniques from [Agarwal et al. 12], [Mayekar et al. 20].
- We construct "difficult" oracles for optimization, compression.
- For $p \in [1, 2)$, the same oracle is "difficult" for optimization, compression.
- For $p \geq 2$, these two oracles differ:
  - The difficult optimization oracle leads to the $\log d$ bound.

Lower Bound

Theorem

1. For $1 \le p < 2$,
$$r^*(T,p) \gtrsim d.$$
2. For $2 \le p$,
$$r^*(T,p) \gtrsim d^{\frac{2}{p}} \vee \log d.$$

- Techniques from [Agarwal et al. 12], [Mayekar et al. 20].
- We construct "difficult" oracles for optimization, compression.
- For $p \in [1,2)$, the same oracle is "difficult" for optimization, compression.
- For $p \ge 2$, these two oracles differ:
  - The difficult optimization oracle leads to the $\log d$ bound.
  - The difficult compression oracle gives the $d^{2/p}$ bound.

Convergence with compressed gradients

Theorem

Consider an unbiased quantizer Q. Then, $\exists$ algorithm π such that

$$\mathcal{E}(T, \infty, p) \cdot \left(\frac{\alpha(Q; p)}{B} \right) \geq \sup_{(f,O) \in \mathcal{O}_p} \mathcal{E}(f, \pi^{QO}, p).$$

Convergence with compressed gradients

Theorem

Consider an unbiased quantizer Q. Then, $\exists$ algorithm π such that

$$\mathcal{E}(T, \infty, p) \cdot \left(\frac{\alpha(Q; p)}{B} \right) \geq \sup_{(f,O) \in \mathcal{O}_p} \mathcal{E}(f, \pi^{QO}, p).$$

$$\alpha(Q; p) \triangleq \sup_{Y \in \mathbb{R}^d : \|Y\|_q^2 \leq B^2 \text{ a.s.}} \sqrt{\mathbb{E}\left[\|Q(Y)\|_q^2 \right]}, \quad p \in [1, 2).$$

$$\alpha(Q; p) \triangleq \sup_{Y \in \mathbb{R}^d : \|Y\|_q^2 \leq B^2 \text{ a.s.}} \sqrt{\mathbb{E}\left[\|Q(Y)\|_2^2 \right]}, \quad p \in [2, \infty].$$

Convergence with compressed gradients

Theorem

Consider an unbiased quantizer Q. Then, $\exists$ algorithm π such that

$$\mathcal{E}(T, \infty, p) \cdot \left(\frac{\alpha(Q;p)}{B} \right) \geq \sup_{(f,O) \in \mathcal{O}_p} \mathcal{E}(f, \pi^{QO}, p).$$

$$\alpha(Q;p) \triangleq \sup_{Y \in \mathbb{R}^d : \|Y\|_q^2 \leq B^2 \text{ a.s.}} \sqrt{\mathbb{E}\left[\|Q(Y)\|_q^2\right]}, \quad p \in [1,2).$$

$$\alpha(Q;p) \triangleq \sup_{Y \in \mathbb{R}^d : \|Y\|_q^2 \leq B^2 \text{ a.s.}} \sqrt{\mathbb{E}\left[\|Q(Y)\|_2^2\right]}, \quad p \in [2,\infty].$$

Design Q such that : 1) Unbiased; 2) $\alpha(Q;p)$ is $O(B)$;

3a) Precision is $O\left(d^{2/p} \vee \log d\right)$ for $p \in [2,\infty]$;

3b) Precision is $O(d)$ for $p \in [1,2)$.

Achievability for $p \in [1, 2)$

Quantizer for $p \in [1, 2)$

Input Y such that $\|Y\|_q \leq B$.

Quantizer for $p \in [1, 2)$

Input Y such that $\|Y\|_q \leq B$.

Split Y such that
$Y_1 := \sum_{i=1}^d Y(i)\mathbb{1}_{\{|Y(i)| \leq c\}} e_i, \quad Y_2 := \sum_{i=1}^d Y(i)\mathbb{1}_{\{|Y(i)| > c\}} e_i,$

where $c = O\left(\frac{B \log\left(d^{1/2-1/q}\right)^{1/q}}{d^{1/q}}\right)$.

Quantizer for $p \in [1, 2)$

Input Y such that $\|Y\|_q \le B$.

Split Y such that
$Y_1 := \sum_{i=1}^d Y(i)\mathbb{1}_{\{|Y(i)| \le c\}} e_i, \quad Y_2 := \sum_{i=1}^d Y(i)\mathbb{1}_{\{|Y(i)| > c\}} e_i,$

where $c = O\left(\frac{B \log\left(d^{1/2-1/q}\right)^{1/q}}{d^{1/q}}\right)$.

Y_1 has small infinity norm; so, a uniform quantizer is good enough.

Quantizer for $p \in [1, 2)$

Input Y such that $\|Y\|_q \leq B$.

Split Y such that
$Y_1 := \sum_{i=1}^{d} Y(i)\mathbb{1}_{\{|Y(i)| \leq c\}} e_i, \quad Y_2 := \sum_{i=1}^{d} Y(i)\mathbb{1}_{\{|Y(i)| > c\}} e_i,$

where $c = O\left(\frac{B \log\left(d^{1/2-1/q}\right)^{1/q}}{d^{1/q}}\right)$.

Y_1 has small infinity norm; so, a uniform quantizer is good enough.

Y_2 is sparse; so, an efficient quantizer for ℓ_2 norm is good enough.

Quantizer for $p \in [1, 2)$

Input Y such that $\|Y\|_q \leq B$.

Split Y such that
$Y_1 := \sum_{i=1}^{d} Y(i) \mathbb{1}_{\{|Y(i)| \leq c\}} e_i, \quad Y_2 := \sum_{i=1}^{d} Y(i) \mathbb{1}_{\{|Y(i)| > c\}} e_i,$

where $c = O\left(\frac{B \log\left(d^{1/2-1/q}\right)^{1/q}}{d^{1/q}} \right)$.

Y_1 has small infinity norm; so, a uniform quantizer is good enough.

Y_2 is sparse; so, an efficient quantizer for ℓ_2 norm is good enough.

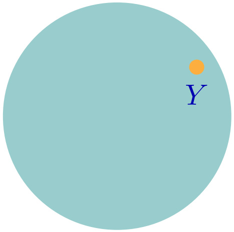
Theorem

$\mathbb{E}\left[Q(Y)|Y\right] = Y; \quad \alpha(Q, p) \leq 4B;$
Precision is $O\left(d + \frac{d}{q} \log\log(d^{1/2-1/q})\right)$ *bits.*

Achievability for $p \in [2, \infty]$

Our Quantizer $SimQ$

Input Y such that $\|Y\|_q \leq B$

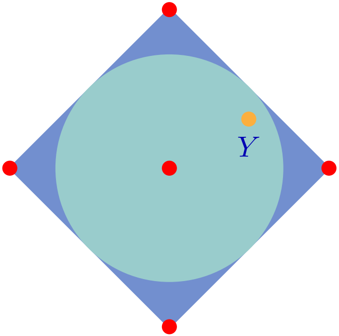


Our Quantizer $SimQ$

Input Y such that $\|Y\|_q \leq B \Rightarrow \|Y\|_1 \leq Bd^{1/p}$.

Encoder

- Sample an i from the set $\{0\} \cup [d]$ with a pmf P, where
  - $\forall i \in [d]$, $P(i) = |Y(i)|/Bd^{1/p}$
  - $P(0) = 1 - \|Y\|_1 / Bd^{1/p}$

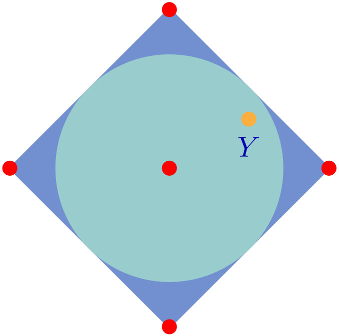


Our Quantizer $SimQ$

Input Y such that $\|Y\|_q \leq B \Rightarrow \|Y\|_1 \leq Bd^{1/p}$.

Encoder

- Sample an i from the set $\{0\} \cup [d]$ with a pmf P, where
  - $\forall i \in [d]$, $P(i) = |Y(i)|/Bd^{1/p}$
  - $P(0) = 1 - \|Y\|_1 / Bd^{1/p}$
- Send i and sign of $Y(i)$.



Our Quantizer $SimQ$

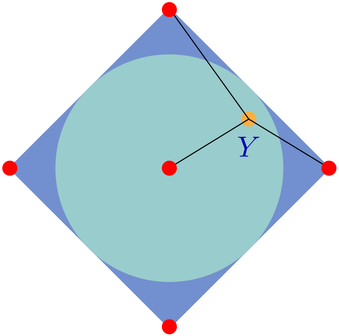
Input Y such that $\|Y\|_q \leq B \Rightarrow \|Y\|_1 \leq Bd^{1/p}$.

Encoder

- Sample an i from the set $\{0\} \cup [d]$ with a pmf P, where
  - $\forall i \in [d]$, $P(i) = |Y(i)|/Bd^{1/p}$
  - $P(0) = 1 - \|Y\|_1 / Bd^{1/p}$
- Send i and sign of $Y(i)$.

Decoder

- Output $Bd^{1/p} \cdot sign(Y(i)) \cdot e_i$



Our Quantizer $SimQ$

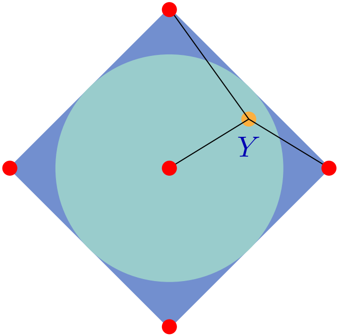
Input Y such that $\|Y\|_q \leq B \Rightarrow \|Y\|_1 \leq Bd^{1/p}$.

Encoder

- Sample an i from the set $\{0\} \cup [d]$ with a pmf P, where
  - $\forall i \in [d]$, $P(i) = |Y(i)|/Bd^{1/p}$
  - $P(0) = 1 - \|Y\|_1 / Bd^{1/p}$
- Send i and sign of $Y(i)$.

Decoder

- Output $Bd^{1/p} \cdot sign(Y(i)) \cdot e_i$



Theorem

$\mathbb{E}[Q(Y)|Y] = Y$; *Precision is* $\log(2d+1)$ *bits;* $\alpha(Q,p) = Bd^{1/p}$.

Our Quantizer $SimQ^+$

- Apply $SimQ$ k times.
- Output the average of k outputs of $SimQ$.

Our Quantizer $SimQ^+$

- Apply $SimQ$ k times.
- Output the average of k outputs of $SimQ$.
- *(Compression step)* Represent the vector of indices using its type.

Our Quantizer $SimQ^+$

- Apply $SimQ$ k times.
- Output the average of k outputs of $SimQ$.
- *(Compression step)* Represent the vector of indices using its type.

Theorem

$\mathbb{E}\left[Q(Y)|Y\right] = Y$; *Precision is* $k \log e + k \log(\frac{d}{k} + 1) + k$ *bits;*
$\alpha(Q,p) \leq \sqrt{\frac{B^2 d^{\frac{2}{p}}}{k} + B^2}$.

Our Quantizer $SimQ^{+}$

- Apply $SimQ$ k times.
- Output the average of k outputs of $SimQ$.
- *(Compression step)* Represent the vector of indices using its type.

Theorem

$\mathbb{E}\left[Q(Y)|Y\right] = Y$; *Precision is* $k \log e + k \log(\frac{d}{k} + 1) + k$ *bits;*
$\alpha(Q,p) \leq \sqrt{\frac{B^2 d^{\frac{2}{p}}}{k} + B^2}$.

By choosing $k = d^{\frac{2}{p}}$, we get $SimQ^{+}$ to be optimal for $p = 2, \infty$.

In Conclusion

Theorem

1. *For* $1 \le p < 2$,
$$r^*(T,p) = \tilde{\Theta}(d).$$
Similar to vector quantization: one bit per dim is needed
2. *For* $2 \le p$,
$$d^{\frac{2}{p}} \vee \log d \lesssim r^*(T,p) \lesssim d^{\frac{2}{p}} \log(d^{1-\frac{2}{p}} + 1).$$
Different from classical vector quantization problem!

Thank You!