Locally Accelerated Conditional Gradients

Jelena Diakonikolas $^1$, Alejandro Carderera $^2$, Sebastian Pokutta $^3$, $^4$

$^1$UC Berkeley, $^2$Georgia Institute of Technology, $^3$Zuse Institute Berlin, $^4$Technische Universität Berlin

AISTATS 2020
Goal is $L$-smooth $\mu$-strongly convex optimization over polytope $\mathcal{X}$.

$$\min_{x \in \mathcal{X}} f(x)$$
Goal is $L$-smooth $\mu$-strongly convex optimization over polytope $\mathcal{X}$.

$$\min_{x \in \mathcal{X}} f(x)$$

Main ingredients:

**First-order (FO) oracle.** Given $x \in \mathcal{X}$ and a differentiable convex function $f : \mathbb{R}^n \to \mathbb{R}$, return:

$$\nabla f(x) \in \mathbb{R}^n \text{ and } f(x) \in \mathbb{R}$$

**Linear optimization (LO) oracle.** Given $v \in \mathbb{R}^n$, return:

$$\arg\min_{x \in \mathcal{X}} \langle v, x \rangle$$
Goal is $L$-smooth $\mu$-strongly convex optimization over polytope $\mathcal{X}$.

$$\min_{x \in \mathcal{X}} f(x)$$

Main ingredients:

**First-order (FO) oracle.** Given $x \in \mathcal{X}$ and a differentiable convex function $f : \mathbb{R}^n \to \mathbb{R}$, return:

$$\nabla f(x) \in \mathbb{R}^n \text{ and } f(x) \in \mathbb{R}$$

**Linear optimization (LO) oracle.** Given $v \in \mathbb{R}^n$, return:

$$\arg\min_{x \in \mathcal{X}} \langle v, x \rangle$$

Focus on *Conditional Gradients/Frank-Wolfe* algorithm [FW56; Pol74] and its variants such as the *Away-step Conditional Gradients/Frank-Wolfe* (AFW) algorithm [Wol70; GM86].
Choose direction that guarantees more progress:

1. **Frank-Wolfe direction:**
   \[
   \argmin_{y \in \mathcal{X}} \langle \nabla f(x), y \rangle - x.
   \]

2. **Away-step direction:**
   \[
   x - \argmax_{y \in \mathcal{S}} \langle \nabla f(x), y \rangle,
   \]
   where \( \mathcal{S} \) is the active set of \( x \).
Convergence rate for \( L \)-smooth \( \mu \)-strongly convex function \( f \)

**Theorem (Convergence rate of AFW)**

[LJ15] Suppose that \( f \) is \( L \)-smooth \( \mu \)-strongly convex over a polytope \( \mathcal{X} \), the number of steps \( T \) required to reach an \( \epsilon \)-optimal solution to the minimization problem satisfies,

\[
T = \mathcal{O} \left( \frac{L}{\mu} \left( \frac{D}{\delta} \right)^2 \log \frac{1}{\epsilon} \right),
\]

where \( D \) and \( \delta \) are the diameter and pyramidal width of \( \mathcal{X} \).
However, we know that optimal methods for this class of functions achieve an $\epsilon$ solution in $T = \mathcal{O}\left(\sqrt{\frac{L}{\mu}} \log \frac{1}{\epsilon}\right)$ first-order calls [NY83; Nes83].

Can CG achieve these convergence rates **globally**?
However, we know that optimal methods for this class of functions achieve an $\epsilon$ solution in $T = \mathcal{O} \left( \sqrt{\frac{L}{\mu}} \log \frac{1}{\epsilon} \right)$ first-order calls [NY83; Nes83].

Can CG achieve these convergence rates **globally**?

*Dimension independent global acceleration is not possible* [Jag13; Lan13].
Objectives:

- Dimension independent global acceleration.
Objectives:

- Dimension independent global acceleration.
- Dimension independent local acceleration.
Locally Accelerated Conditional Gradients (LaCG)

What do we mean by local acceleration?

After a constant number of iterations that does not depend on $\epsilon$, accelerate the convergence.
Let $S_t$ denote the CG active set at iteration $t$.

**What we know:**
\[
\exists r > 0 \text{ s.t. if } \|x^* - x_T\| \leq r \Rightarrow x^* \in \text{conv}(S_T).
\]

Naive Idea: Run an accelerated first-order method (AGD) on $\text{conv}(S_T)$.
We would want the following:
We would want the following:

Problem: The value of $r$ is not known, we don’t know when to switch from AFW to AGD.
We would want the following:

Problem: The value of $r$ is not known, we don’t know when to switch from AFW to AGD.

Challenge: Create algorithm that accelerates without knowledge of $r$. 
Run AFW and restart AGD by running it over a new conv ($S_t$) every $H$ iterations.

Trajectory AFW iterates

- Restart
Run AFW and restart AGD by running it over a new conv ($S_t$) every $H$ iterations.

- Every $H$ iterations restart AGD and run it over conv ($S_t$).
Run AFW and restart AGD by running it over a new conv \((S_t)\) every \(H\) iterations.

- Every \(H\) iterations restart AGD and run it over conv \((S_t)\).
- Have AGD and AFW compete for progress at each iteration between restarts.
Run AFW and restart AGD by running it over a new conv \( (S_t) \) every \( H \) iterations.

- Every \( H \) iterations restart AGD and run it over conv \( (S_t) \).
- Have AGD and AFW compete for progress at each iteration between restarts.
- Space out restarts so that you only lose a factor of 2 in the AGD convergence rate.
What we will obtain:

- AFW-driven convergence
- AGD-driven convergence
- Restart
Algorithm 1 Locally Accelerated Conditional Gradients

1: Initialize $C_0 = S_0$, $x_0 = x_0^{AFW} = x_0^{AGD}$, $H = O \left( \sqrt{\frac{L}{\mu} \log \frac{L}{\mu}} \right)$
2: for $t = 1$ to $T$ do
3: \hspace{1em} $x_{t+1}^{AFW}, S_{t+1} \leftarrow AFW(x_t^{AFW}, S_t)$ \hspace{1em} \text{▷ AFW step}
4: \hspace{1em} if Vertex has been added to $S$ since restart then
5: \hspace{2em} if $t = Hn$ for some $n \in \mathbb{N}$ then
6: \hspace{3em} $x_{t+1}^{AGD} \leftarrow \arg\min_{x \in \{x_t^{AFW}, x_t^{AGD}\}} f(x)$ \hspace{1em} \text{▷ Restart AGD}
7: \hspace{2em} else
8: \hspace{3em} $x_{t+1}^{AGD} \leftarrow AGD(x_t^{AGD}, C_t)$ \hspace{1em} \text{▷ Run AGD decoupled from AFW}
9: \hspace{3em} $C_{t+1} \leftarrow C_t$
10: \hspace{2em} end if
11: \hspace{1em} else
12: \hspace{2em} $x_{t+1}^{AGD} \leftarrow AGD(x_t, C_t)$ \hspace{1em} \text{▷ Run AGD coupled with AFW}
13: \hspace{2em} $C_{t+1} \leftarrow \text{conv}(S_{t+1})$
14: \hspace{2em} end if
15: \hspace{1em} end if
16: \hspace{1em} $x_{t+1} \leftarrow \arg\min_{x \in \{x_{t+1}^{AFW}, x_{t+1}^{AGD}, x_t\}} f(x)$ \hspace{1em} \text{▷ Monotonicity}
17: end for
Analysis relies on the *Approximate Duality Gap* technique [DO19] and the AGD algorithm used is a *Modified μAGD+* algorithm [CDO18; DCP19].

**Theorem (Convergence rate of μAGD+.)**

Let $f$ be $L$-smooth and $μ$-strongly convex and let $\{C_i\}_{i=0}^t$ be a sequence of convex subsets of $\mathcal{X}$ such that $C_i \subseteq C_{i-1}$ for all $i$ and $x^* \in \bigcap_{i=0}^t C_i$, then the $μAGD+$ achieves an $\epsilon$-optimal solution in a number of iterations $T$ that satisfies:

$$T = O \left( \sqrt{\frac{L}{\mu} \log \frac{1}{\epsilon}} \right)$$
Theorem (Convergence rate of LaCG)

Let $f$ be $L$-smooth and $\mu$-strongly convex and let $r$ be the critical radius. The number of steps $T$ required to reach an $\epsilon$-optimal solution to the minimization problem satisfies:

$$t = \min \left\{ \mathcal{O} \left( \frac{L}{\mu} \left( \frac{D}{\delta} \right)^2 \log \frac{1}{\epsilon} \right), K + \mathcal{O} \left( \sqrt{\frac{L}{\mu}} \log \frac{1}{\epsilon} \right) \right\},$$

where $K = \frac{8L}{\mu} \left( \frac{D}{\delta} \right)^2 \log \left( \frac{2(f(x_0) - f^*)}{\mu r^2} \right)$.
Theorem (Convergence rate of LaCG)

Let $f$ be $L$-smooth and $\mu$-strongly convex and let $r$ be the critical radius. The number of steps $T$ required to reach an $\epsilon$-optimal solution to the minimization problem satisfies:

$$t = \min \left\{ O \left( \frac{L}{\mu} \left( \frac{D}{\delta} \right)^2 \log \frac{1}{\epsilon} \right), K + O \left( \sqrt{\frac{L}{\mu}} \log \frac{1}{\epsilon} \right) \right\},$$

where $K = \frac{8L}{\mu} \left( \frac{D}{\delta} \right)^2 \log \left( \frac{2(f(x_0) - f^*)}{\mu r^2} \right)$.

AFW-driven convergence
AGD-driven convergence
Restart
Despite the faster convergence rate after the burn-in phase, how does LaCG perform with respect to other projection-free algorithms?
Simplex in $\mathbb{R}^{1500}$ with $L/\mu = 1000$

Figure: Primal gap vs. iteration

When close enough to $x^*$ (after burn-in phase), there is a significant speedup in the convergence rate.
Birkhoff polytope in $\mathbb{R}^{400\times400}$ with $L/\mu = 100$

Figure: Primal gap vs. iteration

Figure: Primal gap vs. time
Structured Regression over MIPLIB Polytope (ran14x18-disj-8)

Figure: Primal gap vs. iteration

Figure: Primal gap vs. time
Thank you for your attention.
References I


References IV


Can CG achieve these convergence rates **globally**?

Example ([Lan13; Jag13] \( f(x) = \|x\|^2 \) over unit simplex in \( \mathbb{R}^n \).)

We know the optimal solution is given by \( x^* = \mathbb{1}/n \). CG can incorporate at most one vertex in each iteration, if we start from a vertex \( x_0 \), in iteration \( t < n \) we have that:

\[
f(x_t) - f(x^*) \geq \frac{1}{t} - \frac{1}{n}.
\]
Considering iterations such that $t \leq \lfloor n/2 \rfloor$ and rearranging into a linear convergence contraction we have:

$$T = \Omega \left( \frac{1}{r} \log \frac{1}{\varepsilon} \right),$$

where $r \leq 2 \frac{\log 2 t}{2 t}$. 
Considering iterations such that \( t \leq \lceil n/2 \rceil \) and rearranging into a linear convergence contraction we have:

\[
T = \Omega \left( \frac{1}{r} \log \frac{1}{\epsilon} \right),
\]

where \( r \leq 2 \frac{\log 2t}{2t} \).

**Convergence rate of the CG variants for this problem instance:** \( r = \frac{1}{4t} \).

At best a global logarithmic improvement in the convergence rate, therefore **global acceleration in Nesterov’s sense is not possible.**
Other Acceleration Approaches

**Conditional Gradient Sliding (CGS):** Run Nesterov’s Accelerated Gradient Descent, use CG to solve the projection subproblems approximately [LZ16].
Other Acceleration Approaches

**Conditional Gradient Sliding (CGS):** Run Nesterov’s Accelerated Gradient Descent, use CG to solve the projection subproblems approximately [LZ16].

**Catalyst Augmented AFW:** Run Accelerated Proximal Method and solve proximal problems with a linearly convergent CG [LMH15].
Other Acceleration Approaches

**Conditional Gradient Sliding (CGS):** Run Nesterov’s Accelerated Gradient Descent, use CG to solve the projection subproblems approximately [LZ16].

**Catalyst Augmented AFW:** Run Accelerated Proximal Method and solve proximal problems with a linearly convergent CG [LMH15].

**Complexity for** \(L\)-smooth \(\mu\)-strongly convex \(f\).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>LO Calls</th>
<th>FO Calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGS</td>
<td>(\mathcal{O}\left(\frac{LD^2}{\epsilon}\right))</td>
<td>(\mathcal{O}\left(\sqrt{\frac{L}{\mu}} \log \frac{1}{\epsilon}\right))</td>
</tr>
<tr>
<td>Catalyst</td>
<td>(\mathcal{O}\left(\sqrt{\frac{L-\mu}{\mu}} \left(\frac{D}{\delta}\right)^2 \log \frac{1}{\epsilon}\right))</td>
<td>(\mathcal{O}\left(\sqrt{\frac{L-\mu}{\mu}} \left(\frac{D}{\delta}\right)^2 \log \frac{1}{\epsilon}\right))</td>
</tr>
</tbody>
</table>
Congestion Balancing in Traffic Networks

**Figure:** Primal gap vs. iteration

**Figure:** Primal gap vs. time