Distributed Online Frank-Wolfe under Delayed Feedback

Tuan-Anh Nguyen

Journée de Recherche en Apprentissage Frugal
Grenoble, France
\[
\min_{x \in \mathcal{X}} f(x) = \mathbb{E}_{a,b} \left[ \ell(h_x(a), b) \right]
\]
Player's goal: Determine a sequence of actions $x_1, \ldots, x_T$ minimising the cumulative loss $\sum_{t=1}^{T} f_t(x_t)$.
Why Frank-Wolfe?

Vanilla Frank-Wolfe:

1. Linear Oracle: $s_t = \arg\min_{s \in \mathcal{K}} \langle \nabla f(x_t), s \rangle$

2. Update: $x_{t+1} = x_t + \eta_t (s_t - x_t)$

Gradient Descent:

1. Update: $y_{t+1} = x_t - \eta \nabla f(x_t)$

2. Projection: $x_{t+1} = \Pi_{\mathcal{K}}(y_{t+1})$

## Why Frank-Wolfe?

<table>
<thead>
<tr>
<th>Set</th>
<th>Linear minimization</th>
<th>Projection</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$-dimensional $\ell_p$-ball, $p \neq 1, 2, \infty$</td>
<td>$O(n)$</td>
<td>$\tilde{O}(n/\varepsilon^2)$</td>
</tr>
<tr>
<td>Nuclear norm ball of $n \times m$ matrices</td>
<td>$O(n \ln(m + n) \sqrt{\sigma_1}/\sqrt{\varepsilon})$</td>
<td>$O(mn \min{m, n})$</td>
</tr>
<tr>
<td>Flow polytope on a graph with $m$ vertices and $n$ edges with capacity bound on edges</td>
<td>$O((n \log m)(n + m \log m))$</td>
<td>$O(n^4 \log n)$</td>
</tr>
<tr>
<td>Birkhoff polytope ($n \times n$ doubly stochastic matrices)</td>
<td>$O(n^3)$</td>
<td>$\tilde{O}(n^2/\varepsilon^2)$</td>
</tr>
</tbody>
</table>

---

Vanilla Frank-Wolfe:

1. Linear Oracle: \( s_t = \arg\min_{s \in \mathcal{K}} \langle \nabla f(x_t), s \rangle \)

2. Update: \( x_{t+1} = x_t + \eta_t (s_t - x_t) \)

Online Linear Oracle \( \mathcal{O} \):

Sequence of linear loss function \( \langle g_1, \cdot \rangle, \langle g_2, \cdot \rangle, \ldots \)

\[
s_t = \arg\min_{s \in \mathcal{K}} \left\{ \zeta \sum_{l=1}^{t-1} \langle g_l, s \rangle + \langle u, s \rangle \right\}
\]
Delay Mechanism

\[ F_t = \{ s \leq t; s + d_s - 1 = t \} \text{ (or } F_t = \emptyset) \]

\[ F^i_t = \{ s \leq t; s + d^i_s - 1 = t \}, \forall i \in [n] \]

Regret:

\[ R_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} f_t(x) \]

Figure: Given a time \( t \), each agent holds a distinct pool of available gradient feedback that is ready for computation at the current time.
**Centralized Algorithm**

**For some round** $t$

**Prediction**

For $K$ rounds, do

$$s_k \in \mathcal{O}_k$$

$$x_{k+1} = x_k + \eta_k (s_k - x_k)$$

**Update**

For $K$ rounds, do

$$g_k = \sum_{s \in \mathcal{F}_t} \nabla f_s(x_{s,k})$$

**Follow the Perturbed Leader**

$$h_{t-1,k} = \xi \sum_{l=1}^{t-1} \langle g_{l,k}, s \rangle + \langle n, s \rangle$$

$$s_k = \arg\min_{s \in \mathcal{X}} h_{t-1,k}$$

$$h_{t-1,k} + \xi \langle g_k, \cdot \rangle$$

Play $x_t = x_{K+1}$ and receives $\mathcal{F}_t$
For some round $t$

at agent $i$

**Prediction**

For $K$ rounds, do

$s_{i,k} \in \mathcal{O}_{i,k}$

$x_{i,k+1} = y_{i,k} + \eta_k (s_{i,k} - y_{i,k})$

Play $x_{i,t} = x_{i,K+1}$ and receives $F^i_t$

**Update**

For $K$ rounds, do

Surrogate gradient $g_{i,k+1}$ (1)

Local gradient average (2)

Update $\mathcal{O}_{i,k}$ with $d_{i,k}$

(1) $\sum_{s \in F^i_t} \left[ \nabla f_{i,s}(x^i_{s,k+1}) - f_{i,s}(x^i_{s,k}) \right] + d_{i,k}$

(2) $d_{i,k} = \sum_{j=1}^{n} w_{ij} g_{j,k}$
Some comments:

- $K$ Online Linear Oracles $\mathcal{O}_1, \ldots, \mathcal{O}_K$ throughout the learning process.
- Oracles provide estimations of the upcoming gradients’ direction from feedbacks on previous rounds.
- Oracles receive delayed feedback from the algorithms.
- Mixed delayed feedbacks from neighbouring agents in distributed setting.

Impact of delayed feedback to the oracle’s output.
- $s_t$: oracle’s output with delayed feedback

- $\hat{s}_t$: oracle’s output without delayed feedback

\[ \| s_t - \hat{s}_t \| = O \left( \zeta \sum_{s < t} I_{\{s + d_s > t\}} \right) \]

\[ \| s^i_t - \hat{s}^i_t \| = O \left( \zeta \sqrt{n} \left( \frac{\lambda}{\rho} + 1 \right) \frac{1}{n} \sum_{i=1}^{n} \sum_{s < t} I_{\{s + d_s^i > t\}} \right) \]
Informal Theorem 1:

Given $\zeta = \frac{1}{G\sqrt{B}}$, $\eta_k = \min \left(1, \frac{A}{k}\right)$, $K = \sqrt{T}$

$$R_T = O \left( DG\sqrt{B} + R_{T,\emptyset} \right)$$

Additional term related to delayed feedback

Regret of the oracle

$B = \sum_{t=1}^{T} d_t$, sum of all delay value over $T$ rounds
Informal Theorem 2:

Given \( \zeta = \frac{1}{G\sqrt{B}} \), \( \eta_k = \min\left(1, \frac{A}{k}\right) \), \( K = \sqrt{T} \)

\[
R_T = O\left(\sqrt{nDG} \left(\frac{\lambda}{\rho} + 1\right) \sqrt{B} + R_{T,\mathcal{C}}\right)
\]

\( B = \sum_{t=1}^{T} \frac{1}{n} \sum_{i=1}^{n} d_{i,t} \), sum of average delay values over \( n \) agents


Table 1: Comparisons to previous algorithms DGD [Quanrud and Khashabi, 2015] and DOFW [Wan et al., 2022] on centralized online convex optimization with delays bounded by $d$. Our algorithms are in bold.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Centralized</th>
<th>Distributed</th>
<th>Adversarial Delay</th>
<th>Projection-free</th>
<th>Regret</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGD</td>
<td>Yes</td>
<td>-</td>
<td>Yes</td>
<td>-</td>
<td>$O(\sqrt{dT})$</td>
</tr>
<tr>
<td>DOFW</td>
<td>Yes</td>
<td>-</td>
<td>Yes</td>
<td>Yes</td>
<td>$O(T^{3/4} + dT^{1/4})$</td>
</tr>
<tr>
<td>DeLMFW</td>
<td>Yes</td>
<td>-</td>
<td>Yes</td>
<td>Yes</td>
<td>$O(\sqrt{dT})$</td>
</tr>
<tr>
<td>De2MFW</td>
<td>-</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>$O(\sqrt{dT})$</td>
</tr>
</tbody>
</table>
DeLMFW

\[ d = 21 \]
\[ d = 101 \]
$n = 30$

<table>
<thead>
<tr>
<th>$f$</th>
<th>Topology</th>
<th>Erdos Renyi</th>
<th>Grid</th>
<th>Complete</th>
<th>Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>809.37</td>
<td>855.62</td>
<td>799.49</td>
<td>925.72</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>820.15 (+1.3%)</td>
<td>852.15 (-0.4%)</td>
<td>798.79 (-0.08%)</td>
<td>932.34 (+0.7%)</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>834.74 (+3.0%)</td>
<td>868.52 (+1.4%)</td>
<td>802.59 (+0.3%)</td>
<td>971.24 (+4.7%)</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>838.74 (+3.5%)</td>
<td>878.04 (+2.5%)</td>
<td>792.45 (-0.8%)</td>
<td>983.89 (+6.0%)</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>850.49 (+4.9%)</td>
<td>902.30 (+5.3%)</td>
<td>812.21 (+1.5%)</td>
<td>1119.24 (+18.9%)</td>
</tr>
</tbody>
</table>
Positive Results:

- Distributed projection-free algorithm that handling delayed feedback
- Optimal Regret Bound in delay and non-delay setting

Limitation:

- Excessive gradient computation => high communication
Thank you
Follow the Perturbed Leader

**Algorithm 17** FPL for linear losses

1. Input: $\eta > 0$, distribution $\mathcal{D}$ over $\mathbb{R}^n$, decision set $\mathcal{K} \subseteq \mathbb{R}^n$.
2. Sample $\mathbf{n}_0 \sim \mathcal{D}$. Let $\hat{x}_1 = \arg \min_{x \in \mathcal{K}} \{-\mathbf{n}_0^T x\}$.
3. for $t = 1$ to $T$ do
4. Predict $\hat{x}_t$.
5. Observe the linear loss function, suffer loss $\mathbf{g}_t^T \hat{x}_t$.
6. Update

$$\hat{x}_t = \arg \min_{x \in \mathcal{K}} \left\{ \eta \sum_{s=1}^{t-1} \mathbf{g}_s^T x + \mathbf{n}_0^T x \right\}$$

7. end for

**Lemma 1** (Theorem 5.8 [Hazan, 2016]). Given a sequence of linear loss function $f_1, \ldots, f_T$. Suppose that Assumptions 1 to 3 hold true. Let $\mathcal{D}$ be a the uniform distribution over hypercube $[0, 1]^m$. The regret of FTPL is

$$\mathcal{R}_{T,0} \leq \zeta D G^2 T + \frac{1}{\zeta} \sqrt{mD}$$

where $\zeta$ is learning rate of algorithm.

---