Young European Queueing Theorists Workshop, November 2 – 4, 2022

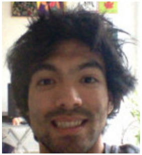
A Tutorial Introduction to (Distributed) Online Convex Optimization

Tim van Erven

UNIVERSITY OF AMSTERDAM

Based on joint work with:

Dirk van der Hoeven



Hedi Hadiji

Example: Electricity Forecasting

- Every day t an electricity company needs to predict how much electricity Y_t is needed the next day
- Given feature vector $\boldsymbol{X}_t \in \mathbb{R}^d$, predict $\hat{Y}_t = \langle \boldsymbol{w}_t, \boldsymbol{X}_t \rangle$ with a linear model
- Next day: observe Y_t
- Measure **loss** by $f_t(\boldsymbol{w}_t) = (Y_t - \hat{Y}_t)^2$ and improve parameter estimates: $\boldsymbol{w}_t \to \boldsymbol{w}_{t+1}$

Example: Electricity Forecasting

- Every day t an electricity company needs to predict how much electricity Y_t is needed the next day
- Given feature vector $\boldsymbol{X}_t \in \mathbb{R}^d$, predict $\hat{Y}_t = \langle \boldsymbol{w}_t, \boldsymbol{X}_t \rangle$ with a linear model
- Next day: observe Y_t
- Measure **loss** by $f_t(\boldsymbol{w}_t) = (Y_t - \hat{Y}_t)^2$ and improve parameter estimates: $\boldsymbol{w}_t \to \boldsymbol{w}_{t+1}$

Goal: Predict almost as well as the best possible parameters $\boldsymbol{u}$:

$$\text{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \sum_{t=1}^{T} f_t(\boldsymbol{u})$$

Online Convex Optimization

Parameters $\boldsymbol{w}$ take values in a convex domain $\mathcal{W} \subset \mathbb{R}^d$

```
for $t = 1, 2, \ldots, T$ do
    Learner predicts $\boldsymbol{w}_t \in \mathcal{W}$
    Nature reveals convex loss function $f_t : \mathcal{W} \to \mathbb{R}$
end for
```

Viewed as a **zero-sum game** against Nature:

$$V \;=\; \min_{\boldsymbol{w}_1} \max_{f_1} \min_{\boldsymbol{w}_2} \max_{f_2} \cdots \min_{\boldsymbol{w}_T} \max_{f_T} \max_{\boldsymbol{u} \in \mathcal{W}} \; \mathrm{Regret}_T(\boldsymbol{u})$$

Online Convex Optimization

```
Parameters $\boldsymbol{w}$ take values in a convex domain $\mathcal{W} \subset \mathbb{R}^d$
for $t = 1, 2, \dots, T$ do
    Learner predicts $\boldsymbol{w}_t \in \mathcal{W}$
    Nature reveals convex loss function $f_t : \mathcal{W} \to \mathbb{R}$
end for
```

Viewed as a **zero-sum game** against Nature:

$$V = \min_{\boldsymbol{w}_1} \max_{f_1} \min_{\boldsymbol{w}_2} \max_{f_2} \cdots \min_{\boldsymbol{w}_T} \max_{f_T} \max_{\boldsymbol{u} \in \mathcal{W}} \text{Regret}_T(\boldsymbol{u})$$

Make standard assumptions:

- Domain $\mathcal{W}$ compact with diameter at most D
- Bounded gradients: $\|\nabla f_t(\boldsymbol{w}_t)\| \leq G$

Online Gradient Descent

$$\begin{aligned} \tilde{\boldsymbol{w}}_{t+1} &= \boldsymbol{w}_t - \eta_t \nabla f_t(\boldsymbol{w}_t) \\ \boldsymbol{w}_{t+1} &= \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathcal{W}} \|\boldsymbol{w} - \tilde{\boldsymbol{w}}_{t+1}\| \end{aligned}$$

Theorem (Zinkevich, 2003)

Online gradient descent with $\eta_t = \frac{D}{G\sqrt{t}}$ guarantees

$$\mathrm{Regret}_T(\boldsymbol{u}) \leq \frac{3}{2} DG\sqrt{T}$$

for **any** *choices of Nature.*

Without further assumptions, this is **optimal** up to the constant factor.
(If T is known in advance, the optimal constant is 1.)

OGD Analysis

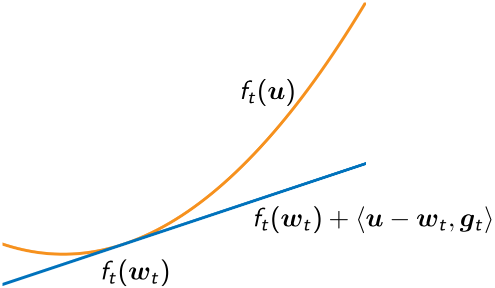
Simplifications: Assume no projections, constant learning rate:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t)$$

Proof:

1. Reduction to Linear Losses

By convexity of f_t, abbreviating $\boldsymbol{g}_t = \nabla f_t(\boldsymbol{w}_t)$:


$f_t(\boldsymbol{u})$
$f_t(\boldsymbol{w}_t) + \langle \boldsymbol{u} - \boldsymbol{w}_t, \boldsymbol{g}_t \rangle$
$f_t(\boldsymbol{w}_t)$

$$\text{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T} \Big(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u}) \Big) \leq \sum_{t=1}^{T} \Big(\langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle - \langle \boldsymbol{u}, \boldsymbol{g}_t \rangle \Big)$$

OGD Analysis

Simplifications: Assume no projections, constant learning rate:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t)$$

Proof:

2. Analyzing Linear Losses, $\boldsymbol{g}_t = \nabla f_t(\boldsymbol{w}_t)$

$$\begin{aligned} \|\boldsymbol{w}_{t+1} - \boldsymbol{u}\|^2 &= \|\boldsymbol{w}_t - \boldsymbol{u} - \eta \boldsymbol{g}_t\|^2 \\ &= \|\boldsymbol{w}_t - \boldsymbol{u}\|^2 - 2\eta \langle \boldsymbol{w}_t - \boldsymbol{u}, \boldsymbol{g}_t \rangle + \eta^2 \|\boldsymbol{g}_t\|^2 \end{aligned}$$

OGD Analysis

Simplifications: Assume no projections, constant learning rate:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t)$$

Proof:

2. Analyzing Linear Losses, $\boldsymbol{g}_t = \nabla f_t(\boldsymbol{w}_t)$

$$\begin{aligned} \|\boldsymbol{w}_{t+1} - \boldsymbol{u}\|^2 &= \|\boldsymbol{w}_t - \boldsymbol{u} - \eta \boldsymbol{g}_t\|^2 \\ &= \|\boldsymbol{w}_t - \boldsymbol{u}\|^2 - 2\eta \langle \boldsymbol{w}_t - \boldsymbol{u}, \boldsymbol{g}_t \rangle + \eta^2 \|\boldsymbol{g}_t\|^2 \\ \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle - \langle \boldsymbol{u}, \boldsymbol{g}_t \rangle &= \frac{1}{2\eta}\|\boldsymbol{w}_t - \boldsymbol{u}\|^2 - \frac{1}{2\eta}\|\boldsymbol{w}_{t+1} - \boldsymbol{u}\|^2 + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2 \end{aligned}$$

OGD Analysis

Simplifications: Assume no projections, constant learning rate:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t)$$

Proof:

2. Analyzing Linear Losses, $\boldsymbol{g}_t = \nabla f_t(\boldsymbol{w}_t)$

$$\begin{aligned}
\|\boldsymbol{w}_{t+1} - \boldsymbol{u}\|^2 &= \|\boldsymbol{w}_t - \boldsymbol{u} - \eta \boldsymbol{g}_t\|^2 \\
&= \|\boldsymbol{w}_t - \boldsymbol{u}\|^2 - 2\eta\langle \boldsymbol{w}_t - \boldsymbol{u}, \boldsymbol{g}_t\rangle + \eta^2\|\boldsymbol{g}_t\|^2 \\
\langle \boldsymbol{w}_t, \boldsymbol{g}_t\rangle - \langle \boldsymbol{u}, \boldsymbol{g}_t\rangle &= \frac{1}{2\eta}\|\boldsymbol{w}_t - \boldsymbol{u}\|^2 - \frac{1}{2\eta}\|\boldsymbol{w}_{t+1} - \boldsymbol{u}\|^2 + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2 \\
\sum_{t=1}^{T}\Big(\langle \boldsymbol{w}_t, \boldsymbol{g}_t\rangle - \langle \boldsymbol{u}, \boldsymbol{g}_t\rangle\Big) &= \frac{1}{2\eta}\|\boldsymbol{w}_1 - \boldsymbol{u}\|^2 - \frac{1}{2\eta}\|\boldsymbol{w}_{T+1} - \boldsymbol{u}\|^2 + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|^2
\end{aligned}$$

OGD Analysis

Simplifications: Assume no projections, constant learning rate:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t)$$

Proof:

2. Analyzing Linear Losses, $\boldsymbol{g}_t = \nabla f_t(\boldsymbol{w}_t)$

$$\begin{aligned}
\|\boldsymbol{w}_{t+1} - \boldsymbol{u}\|^2 &= \|\boldsymbol{w}_t - \boldsymbol{u} - \eta \boldsymbol{g}_t\|^2 \\
&= \|\boldsymbol{w}_t - \boldsymbol{u}\|^2 - 2\eta \langle \boldsymbol{w}_t - \boldsymbol{u}, \boldsymbol{g}_t \rangle + \eta^2 \|\boldsymbol{g}_t\|^2 \\
\langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle - \langle \boldsymbol{u}, \boldsymbol{g}_t \rangle &= \frac{1}{2\eta}\|\boldsymbol{w}_t - \boldsymbol{u}\|^2 - \frac{1}{2\eta}\|\boldsymbol{w}_{t+1} - \boldsymbol{u}\|^2 + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2 \\
\sum_{t=1}^{T} \Big(\langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle - \langle \boldsymbol{u}, \boldsymbol{g}_t \rangle \Big) &= \frac{1}{2\eta}\|\boldsymbol{w}_1 - \boldsymbol{u}\|^2 - \frac{1}{2\eta}\|\boldsymbol{w}_{T+1} - \boldsymbol{u}\|^2 + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|^2 \\
\text{Regret}_T(\boldsymbol{u}) &\leq \frac{1}{2\eta}\|\boldsymbol{w}_1 - \boldsymbol{u}\|^2 + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|^2 \leq \frac{1}{2\eta}D^2 + \frac{\eta}{2}G^2 T
\end{aligned}$$

OGD Analysis

Simplifications: Assume no projections, constant learning rate:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t)$$

Proof:

2. Analyzing Linear Losses, $\boldsymbol{g}_t = \nabla f_t(\boldsymbol{w}_t)$

$$\begin{aligned}
\|\boldsymbol{w}_{t+1} - \boldsymbol{u}\|^2 &= \|\boldsymbol{w}_t - \boldsymbol{u} - \eta \boldsymbol{g}_t\|^2 \\
&= \|\boldsymbol{w}_t - \boldsymbol{u}\|^2 - 2\eta\langle \boldsymbol{w}_t - \boldsymbol{u}, \boldsymbol{g}_t\rangle + \eta^2\|\boldsymbol{g}_t\|^2 \\
\langle \boldsymbol{w}_t, \boldsymbol{g}_t\rangle - \langle \boldsymbol{u}, \boldsymbol{g}_t\rangle &= \frac{1}{2\eta}\|\boldsymbol{w}_t - \boldsymbol{u}\|^2 - \frac{1}{2\eta}\|\boldsymbol{w}_{t+1} - \boldsymbol{u}\|^2 + \frac{\eta}{2}\|\boldsymbol{g}_t\|^2 \\
\sum_{t=1}^{T}\Big(\langle \boldsymbol{w}_t, \boldsymbol{g}_t\rangle - \langle \boldsymbol{u}, \boldsymbol{g}_t\rangle\Big) &= \frac{1}{2\eta}\|\boldsymbol{w}_1 - \boldsymbol{u}\|^2 - \frac{1}{2\eta}\|\boldsymbol{w}_{T+1} - \boldsymbol{u}\|^2 + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|^2 \\
\text{Regret}_T(\boldsymbol{u}) &\le \frac{1}{2\eta}\|\boldsymbol{w}_1 - \boldsymbol{u}\|^2 + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|^2 \le \frac{1}{2\eta}D^2 + \frac{\eta}{2}G^2T \\
&= DG\sqrt{T} \qquad \text{for } \eta = \frac{D}{G\sqrt{T}}
\end{aligned}$$

Online Convex Optimization with Delays

Delayed Feedback:

- Suppose $\boldsymbol{g}_t$ not observed at end of round t, but later
- Let $\mathcal{U}_t \subset \{1, \dots, t-1\}$ list missing gradients at start of round t

Online Convex Optimization with Delays

Delayed Feedback:

- Suppose $\boldsymbol{g}_t$ not observed at end of round t, but later
- Let $\mathcal{U}_t \subset \{1, \ldots, t-1\}$ list missing gradients at start of round t

Theorem (McMahan, Streeter, 2014)

Online gradient descent (without projections and with $\eta_t = \eta$) using only the available gradients guarantees

$$\begin{aligned}
\text{Regret}_T(\boldsymbol{u}) &\leq \frac{1}{2\eta}\|\boldsymbol{w}_1 - \boldsymbol{u}\|^2 + \frac{\eta}{2}\sum_{t=1}^{T}\Big(\|\boldsymbol{g}_t\|^2 + 2\|\boldsymbol{g}_t\|\sum_{s\in\mathcal{U}_t}\|\boldsymbol{g}_s\|\Big)\\
&\leq \frac{1}{2\eta}D + \frac{\eta}{2}(1+2\tau)G^2T \qquad \textit{if } |\mathcal{U}_t| \leq \tau\\
&= DG\sqrt{(1+2\tau)T} \qquad \textit{for } \eta = \frac{D}{G\sqrt{(1+2\tau)T}}
\end{aligned}$$

Delayed Feedback Analysis

1. Reduction to linear losses
2. Regret of OGD with delayed feedback $\boldsymbol{w}_t$ is at most:
   - Regret of oracle OGD $\boldsymbol{w}_t^*$ that observes all gradients
   - $+$ differences in linear losses between $\boldsymbol{w}_t$ and $\boldsymbol{w}_t^*$:

$$\begin{aligned}
&\sum_{t=1}^{T}\Big(\langle \boldsymbol{w}_t, \boldsymbol{g}_t\rangle - \langle \boldsymbol{w}_t^*, \boldsymbol{g}_t\rangle\Big) \\
&\qquad = \sum_{t=1}^{T}\Big(\langle \boldsymbol{w}_1 - \eta \sum_{s \in [t-1] \setminus \mathcal{U}_t} \boldsymbol{g}_s, \boldsymbol{g}_t\rangle - \langle \boldsymbol{w}_1 - \eta \sum_{s \in [t-1]} \boldsymbol{g}_s, \boldsymbol{g}_t\rangle\Big) \\
&\qquad = \sum_{t=1}^{T}\langle \eta \sum_{s \in \mathcal{U}_t} \boldsymbol{g}_s, \boldsymbol{g}_t\rangle \\
&\qquad \leq \eta \sum_{t=1}^{T} \|\boldsymbol{g}_t\| \sum_{s \in \mathcal{U}_t} \|\boldsymbol{g}_s\|
\end{aligned}$$

$$\text{Regret}_T(\boldsymbol{u}) \leq \frac{1}{2\eta}\|\boldsymbol{w}_1 - \boldsymbol{u}\|^2 + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|^2 + \eta \sum_{t=1}^{T}\|\boldsymbol{g}_t\| \sum_{s \in \mathcal{U}_t}\|\boldsymbol{g}_s\|$$

Distributed Online Convex Optimization

[Van der Hoeven, Hadiji, Van Erven, 2022]:

Given **connection graph** $\mathcal{G}$ between N agents:

```
for $t = 1, 2, \dots, T$ do
    Nature activates agent $I_t \in \{1, \dots, N\}$
    Active agent $I_t$ predicts $\boldsymbol{w}_t \in \mathcal{W}$
    Nature reveals convex loss function $f_t : \mathcal{W} \to \mathbb{R}$ only to agent $I_t$
    All agents can send a message to their neighbors in $\mathcal{G}$
end for
```

Agents cooperate to minimize joint regret:

$$\text{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \sum_{t=1}^{T} f_t(\boldsymbol{u})$$

Distributed Learning Causes Delayed Feedback

Incurring the maximum delay:

- If **graph diameter** is $\mathrm{diam}(\mathcal{G})$, then it takes at most $\mathrm{diam}(\mathcal{G})$ rounds to transmit each gradient $\boldsymbol{g}_t$ to all agents
- So each agent can run OGD with feedback delay $\tau = \mathrm{diam}(\mathcal{G})$ to get

$$\mathrm{Regret}_T(\boldsymbol{u}) = O\left(DG\sqrt{\mathrm{diam}(\mathcal{G})T}\right)$$

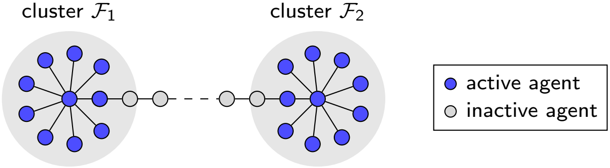
Distributed Learning Causes Delayed Feedback

Incurring the maximum delay:

- If **graph diameter** is $\mathrm{diam}(\mathcal{G})$, then it takes at most $\mathrm{diam}(\mathcal{G})$ rounds to transmit each gradient $\boldsymbol{g}_t$ to all agents
- So each agent can run OGD with feedback delay $\tau = \mathrm{diam}(\mathcal{G})$ to get

$$\mathrm{Regret}_T(\boldsymbol{u}) = O\Big(DG\sqrt{\mathrm{diam}(\mathcal{G})T}\Big)$$

This is suboptimal:



Two clusters that can be made arbitrarily far apart
by extending the line that connects them

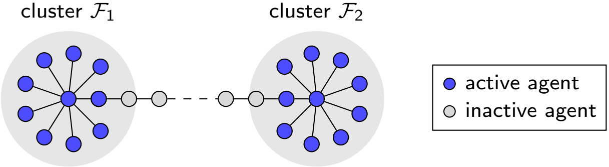
Distributed Learning Causes Delayed Feedback

Incurring the maximum delay:

- If **graph diameter** is $\operatorname{diam}(\mathcal{G})$, then it takes at most $\operatorname{diam}(\mathcal{G})$ rounds to transmit each gradient $\boldsymbol{g}_t$ to all agents
- So each agent can run OGD with feedback delay $\tau = \operatorname{diam}(\mathcal{G})$ to get

$$\mathrm{Regret}_T(\boldsymbol{u}) = O\Big(DG\sqrt{\operatorname{diam}(\mathcal{G})\,T}\Big)$$

This is suboptimal:



Two clusters that can be made arbitrarily far apart by extending the line that connects them

Much better: Learn separately for each cluster:

$$\mathrm{Regret}_T(\boldsymbol{u}) = O\Big(DG\sqrt{\operatorname{diam}(\mathcal{F}_1)\,T} + DG\sqrt{\operatorname{diam}(\mathcal{F}_2)\,T}\Big)$$

But optimal clustering depends on activations. How do we learn it?

Learning the Best Graph Partition

Given collection $\mathcal{Q}$ of subgraphs of $\mathcal{G}$, a **$\mathcal{Q}$-partition** is a partition $\{\mathcal{F}_1, \ldots, \mathcal{F}_r\}$ of $\mathcal{G}$ such that each $\mathcal{F}_i \in \mathcal{Q}$.

Theorem (Van der Hoeven, Hadiji, Van Erven, 2022)

Given any $\mathcal{Q}$, there exists an algorithm that guarantees

$$\sum_{j=1}^{r} \mathrm{Regret}_{\mathcal{F}_j}(\boldsymbol{u}_j) = O\Big(\sum_{j=1}^{r} \|\boldsymbol{u}_j\| G \Big(\sqrt{\mathrm{diam}(\mathcal{F}_j) T_j \ln(1 + |\mathcal{Q}| \, \mathrm{diam}(\mathcal{F}_j) \|\boldsymbol{u}_j\| T_j)}\Big)\Big)$$

for any $\mathcal{Q}$-partition $\{\mathcal{F}_1, \ldots, \mathcal{F}_r\}$ and any $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r \in \mathcal{W}$.

$$\mathrm{Regret}_{\mathcal{F}_j}(\boldsymbol{u}) = \sum_{t : I_t \in \mathcal{F}_j} (f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u}))$$

Comparator-Adaptive Algorithms

Unbounded domain:

- $\text{Regret}_T(\boldsymbol{u}) = O(DG\sqrt{T})$ when comparator $\boldsymbol{u} \in \mathcal{W}$ with diameter of $\mathcal{W}$ at most D.
- What if we have **no bound** a priori on **comparator norm** $\|\boldsymbol{u}\|$, so we want to consider $\mathcal{W} = \mathbb{R}^d$?

Theorem (McMahan, Streeter, 2012)

Given G and any $\epsilon > 0$, there exists an online algorithm that achieves

$$\text{Regret}_T(\boldsymbol{u}) = O(\|\boldsymbol{u}\| G \sqrt{T \log \tfrac{(1+\|\boldsymbol{u}\|)T}{\epsilon}} + \epsilon G) \qquad \textit{for all } \boldsymbol{u} \in \mathbb{R}^d.$$

- Essentially as good as **bounded domain** $\mathcal{W} = \{\boldsymbol{w} : \|\boldsymbol{w}\| \leq \frac{1}{2}D\}$ for **oracle choice** $D = 2\|\boldsymbol{u}\|$.

Aggregating Multiple Online Methods

Aggregation:

- Given K **online learning algorithms** with iterates $\boldsymbol{w}_t^1, \ldots, \boldsymbol{w}_t^K$
- Predict almost as well as the best one k^*:

$$\text{Regret}_T(\boldsymbol{u}) \leq \text{Regret}_T^{k^*}(\boldsymbol{u}) + \text{overhead}$$

Aggregating Multiple Online Methods

Aggregation:

- Given K **online learning algorithms** with iterates $\boldsymbol{w}_t^1, \ldots, \boldsymbol{w}_t^K$
- Predict almost as well as the best one k^*:

$$\text{Regret}_T(\boldsymbol{u}) \leq \text{Regret}_T^{k^*}(\boldsymbol{u}) + \text{overhead}$$

Results: [Littlestone, Warmuth, 1994], [Vovk, 1998]: If $f_t(\boldsymbol{w}_t^k) \in [a, b]$, then can achieve

$$\text{overhead} = O((b-a)\sqrt{T \ln K})$$

Aggregating Multiple Online Methods

Aggregation:

- Given K **online learning algorithms** with iterates $\boldsymbol{w}_t^1, \ldots, \boldsymbol{w}_t^K$
- Predict almost as well as the best one k^*:

$$\text{Regret}_T(\boldsymbol{u}) \leq \text{Regret}_T^{k^*}(\boldsymbol{u}) + \text{overhead}$$

Results: [Littlestone, Warmuth, 1994], [Vovk, 1998]: If $f_t(\boldsymbol{w}_t^k) \in [a, b]$, then can achieve

$$\text{overhead} = O((b-a)\sqrt{T \ln K})$$

[Cuskosky, 2019]: For comparator-adaptive methods with linear(ized) losses, simple iterate addition $\boldsymbol{w}_t = \sum_{k=1}^K \boldsymbol{w}_t^k$ achieves

$$\text{overhead} = \sum_{k \neq k^*} \text{Regret}_T^k(\mathbf{0}) = O(\epsilon K G) \qquad \text{think: } \epsilon \propto 1/K$$

Aggregating Multiple Online Methods

Aggregation:

- Given K **online learning algorithms** with iterates $\boldsymbol{w}_t^1, \ldots, \boldsymbol{w}_t^K$
- Predict almost as well as the best one k^*:

$$\text{Regret}_T(\boldsymbol{u}) \leq \text{Regret}_T^{k^*}(\boldsymbol{u}) + \text{overhead}$$

Results: [Littlestone, Warmuth, 1994], [Vovk, 1998]: If $f_t(\boldsymbol{w}_t^k) \in [a, b]$, then can achieve

$$\text{overhead} = O((b-a)\sqrt{T \ln K})$$

[Cuskosky, 2019]: For comparator-adaptive methods with linear(ized) losses, simple iterate addition $\boldsymbol{w}_t = \sum_{k=1}^K \boldsymbol{w}_t^k$ achieves

$$\text{overhead} = \sum_{k \neq k^*} \text{Regret}_T^k(\boldsymbol{0}) = O(\epsilon K G) \qquad \text{think: } \epsilon \propto 1/K$$

Proof:

$$\sum_{t=1}^T \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle - \langle \boldsymbol{u}, \boldsymbol{g}_t \rangle = \sum_{k=1}^K \sum_{t=1}^T \langle \boldsymbol{w}_t^k, \boldsymbol{g}_t \rangle - \sum_{t=1}^T \langle \boldsymbol{u}, \boldsymbol{g}_t \rangle$$

$$= \sum_{t=1}^T \Big(\langle \boldsymbol{w}_t^{k^*}, \boldsymbol{g}_t \rangle - \langle \boldsymbol{u}, \boldsymbol{g}_t \rangle \Big) + \sum_{k \neq k^*} \sum_{t=1}^T \Big(\langle \boldsymbol{w}_t^k, \boldsymbol{g}_t \rangle - \langle \boldsymbol{0}, \boldsymbol{g}_t \rangle \Big)$$

Learning the Graph Partition: Approach

Challenge:

- For each node i in the graph and cell $\mathcal{F}_j \in \mathcal{Q}$ that contains i, construct an algorithm $\boldsymbol{w}_t^{(i,j)}$ that can **handle delays** $\tau = \mathrm{diam}(\mathcal{F}_j)$
- Then i **aggregates iterates** $\boldsymbol{w}_t^{(i,j)}$ for all such j
- Problem: standard aggregation techniques with delays incur overhead that depends on **maximum delay** $\max_j \mathrm{diam}(\mathcal{F}_j)$

Learning the Graph Partition: Approach

Challenge:

- For each node i in the graph and cell $\mathcal{F}_j \in \mathcal{Q}$ that contains i, construct an algorithm $\boldsymbol{w}_t^{(i,j)}$ that can **handle delays** $\tau = \mathrm{diam}(\mathcal{F}_j)$
- Then i **aggregates iterates** $\boldsymbol{w}_t^{(i,j)}$ for all such j
- Problem: standard aggregation techniques with delays incur overhead that depends on **maximum delay** $\max_j \mathrm{diam}(\mathcal{F}_j)$

Our Solution:

- Make sure that $\boldsymbol{w}_t^{(i,j)}$ not only can handle delays, but are **also comparator adaptive** (new result)
- Then aggregation is possible using **iterate addition**, with overhead that depends on $\mathrm{diam}(\mathcal{F}_j)$ for optimal $\mathcal{F}_j$.
- Project $\boldsymbol{w}_t$ onto bounded $\mathcal{W}$ using black-box reduction by [Cutkosky, Orabona, 2018]

Summary

Online Convex Optimization

- Online gradient descent
- Delayed feedback
- Comparator-adaptive algorithms
- Aggregating multiple online methods
- New: Combined comparator-adaptive + delayed feedback

Distributed Online Convex Optimization

- Agents in a graph cooperate to minimize joint regret
- New: Learning the best graph partition

References

- D. van der Hoeven, H. Hadiji and T. van Erven. **Distributed Online Learning for Joint Regret with Communication Constraints**, Proceedings of the 33rd International Conference on Algorithmic Learning Theory (ALT), no. 167, pp. 1003-1042, 2022.