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 \( X_t \in \mathbb{R}^d \), predict \( \hat{Y}_t = \langle w_t, X_t \rangle \) with a linear model
- Next day: observe \( Y_t \)
- Measure loss by \( f_t(w_t) = (Y_t - \hat{Y}_t)^2 \) and improve parameter estimates: \( w_t \rightarrow w_{t+1} \)
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 $X_t \in \mathbb{R}^d$, predict $\hat{Y}_t = \langle w_t, X_t \rangle$ with a linear model
▶ Next day: observe $Y_t$
▶ Measure loss by $f_t(w_t) = (Y_t - \hat{Y}_t)^2$ and improve parameter estimates: $w_t \rightarrow w_{t+1}$

**Goal:** Predict almost as well as the best possible parameters $u$:

$$\text{Regret}_T(u) = \sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(u)$$
Online Convex Optimization

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

1. for $t = 1, 2, \ldots, T$ do
2. Learner predicts $w_t \in \mathcal{W}$
3. Nature reveals convex loss function $f_t : \mathcal{W} \to \mathbb{R}$
4. end for

Viewed as a zero-sum game against Nature:

\[
V = \min_{w_1} \max_{f_1} \min_{w_2} \max_{f_2} \cdots \min_{w_T} \max_{f_T} \max_{u \in \mathcal{W}} \text{Regret}_T(u)
\]
Online Convex Optimization

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

1: \textbf{for} $t = 1, 2, \ldots, T$ \textbf{do}
2: \hspace{1em} Learner predicts $w_t \in \mathcal{W}$
3: \hspace{1em} Nature reveals convex loss function $f_t : \mathcal{W} \to \mathbb{R}$
4: \textbf{end for}

Viewed as a \textbf{zero-sum game} against Nature:

$$V = \min_{w_1} \max_{f_1} \min_{w_2} \max_{f_2} \cdots \min_{w_T} \max_{f_T} \max_{u \in \mathcal{W}} \text{Regret}_T(u)$$

\textbf{Make standard assumptions:}

- Domain $\mathcal{W}$ compact with diameter at most $D$
- Bounded gradients: $\|\nabla f_t(w_t)\| \leq G$
Online Gradient Descent

\[
\begin{align*}
\tilde{w}_{t+1} &= w_t - \eta_t \nabla f_t(w_t) \\
w_{t+1} &= \arg\min_{w \in \mathcal{W}} \|w - \tilde{w}_{t+1}\|
\end{align*}
\]

**Theorem (Zinkevich, 2003)**

*Online gradient descent with* \(\eta_t = \frac{D}{G \sqrt{t}}\) *guarantees*

\[
\text{Regret}_T(u) \leq \frac{3}{2} DG \sqrt{T}
\]

*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:

\[ w_{t+1} = w_t - \eta \nabla f_t(w_t) \]

Proof:

1. Reduction to Linear Losses

By convexity of \( f_t \), abbreviating \( g_t = \nabla f_t(w_t) \):

\[
\text{Regret}_T(u) = \sum_{t=1}^{T} \left( f_t(w_t) - f_t(u) \right) \leq \sum_{t=1}^{T} \left( \langle w_t, g_t \rangle - \langle u, g_t \rangle \right)
\]
**OGD Analysis**

**Simplifications:** Assume no projections, constant learning rate:

\[ w_{t+1} = w_t - \eta \nabla f_t(w_t) \]

**Proof:**

2. **Analyzing Linear Losses**, \( g_t = \nabla f_t(w_t) \)

\[
\| w_{t+1} - u \|^2 = \| w_t - u - \eta g_t \|^2 \\
= \| w_t - u \|^2 - 2\eta \langle w_t - u, g_t \rangle + \eta^2 \| g_t \|^2
\]
OGD Analysis

**Simplifications:** Assume no projections, constant learning rate:

\[ w_{t+1} = w_t - \eta \nabla f_t(w_t) \]

**Proof:**

2. **Analyzing Linear Losses**, \( g_t = \nabla f_t(w_t) \)

\[
\| w_{t+1} - u \|^2 = \| w_t - u - \eta g_t \|^2 \\
= \| w_t - u \|^2 - 2\eta \langle w_t - u, g_t \rangle + \eta^2 \| g_t \|^2 \\
\langle w_t, g_t \rangle - \langle u, g_t \rangle = \frac{1}{2\eta} \| w_t - u \|^2 - \frac{1}{2\eta} \| w_{t+1} - u \|^2 + \frac{\eta}{2} \| g_t \|^2 \]
OGD Analysis

**Simplifications:** Assume no projections, constant learning rate:

\[ w_{t+1} = w_t - \eta \nabla f_t(w_t) \]

**Proof:**

2. **Analyzing Linear Losses**, \( g_t = \nabla f_t(w_t) \)

\[
||w_{t+1} - u||^2 = ||w_t - u - \eta g_t||^2 \\
= ||w_t - u||^2 - 2\eta \langle w_t - u, g_t \rangle + \eta^2 ||g_t||^2 \\
\langle w_t, g_t \rangle - \langle u, g_t \rangle = \frac{1}{2\eta} ||w_t - u||^2 - \frac{1}{2\eta} ||w_{t+1} - u||^2 + \frac{\eta}{2} ||g_t||^2 \\
\sum_{t=1}^{T} \left( \langle w_t, g_t \rangle - \langle u, g_t \rangle \right) = \frac{1}{2\eta} ||w_1 - u||^2 - \frac{1}{2\eta} ||w_{T+1} - u||^2 + \frac{\eta}{2} \sum_{t=1}^{T} ||g_t||^2 \]
OGD Analysis

Simplifications: Assume no projections, constant learning rate:

\[ w_{t+1} = w_t - \eta \nabla f_t(w_t) \]

Proof:

2. Analyzing Linear Losses, \( g_t = \nabla f_t(w_t) \)

\[ \|w_{t+1} - u\|^2 = \|w_t - u - \eta g_t\|^2 \]
\[ = \|w_t - u\|^2 - 2\eta \langle w_t - u, g_t \rangle + \eta^2 \|g_t\|^2 \]

\[ \langle w_t, g_t \rangle - \langle u, g_t \rangle = \frac{1}{2\eta} \|w_t - u\|^2 - \frac{1}{2\eta} \|w_{t+1} - u\|^2 + \frac{\eta}{2} \|g_t\|^2 \]

\[ \sum_{t=1}^{T} \left( \langle w_t, g_t \rangle - \langle u, g_t \rangle \right) = \frac{1}{2\eta} \|w_1 - u\|^2 - \frac{1}{2\eta} \|w_{T+1} - u\|^2 + \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|^2 \]

\[ \text{Regret}_T(u) \leq \frac{1}{2\eta} \|w_1 - u\|^2 + \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|^2 \leq \frac{1}{2\eta} D^2 + \frac{\eta}{2} G^2 T \]
OGD Analysis

Simplifications: Assume no projections, constant learning rate:

\[ w_{t+1} = w_t - \eta \nabla f_t(w_t) \]

Proof:

2. Analyzing Linear Losses, \( g_t = \nabla f_t(w_t) \)

\[
\|w_{t+1} - u\|^2 = \|w_t - u - \eta g_t\|^2
\]

\[
= \|w_t - u\|^2 - 2\eta \langle w_t - u, g_t \rangle + \eta^2 \|g_t\|^2
\]

\[
\langle w_t, g_t \rangle - \langle u, g_t \rangle = \frac{1}{2\eta} \|w_t - u\|^2 - \frac{1}{2\eta} \|w_{t+1} - u\|^2 + \frac{\eta}{2} \|g_t\|^2
\]

\[
\sum_{t=1}^{T} \left( \langle w_t, g_t \rangle - \langle u, g_t \rangle \right) = \frac{1}{2\eta} \|w_1 - u\|^2 - \frac{1}{2\eta} \|w_{T+1} - u\|^2 + \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|^2
\]

\[
\text{Regret}_T(u) \leq \frac{1}{2\eta} \|w_1 - u\|^2 + \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|^2 \leq \frac{1}{2\eta} D^2 + \frac{\eta}{2} G^2 T
\]

\[
= DG \sqrt{T} \quad \text{for } \eta = \frac{D}{G \sqrt{T}}
\]
Online Convex Optimization with Delays

Delayed Feedback:

- Suppose \( g_t \) 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 \( t \)
Online Convex Optimization with Delays

Delayed Feedback:
- Suppose $g_t$ 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*

\[
\text{Regret}_T(u) \leq \frac{1}{2\eta} \|w_1 - u\|^2 + \frac{\eta}{2} \sum_{t=1}^{T} \left(\|g_t\|^2 + 2\|g_t\| \sum_{s \in \mathcal{U}_t} \|g_s\|\right)
\]

\[
\leq \frac{1}{2\eta} D + \frac{\eta}{2} (1 + 2\tau) G^2 T \quad \text{if } |\mathcal{U}_t| \leq \tau
\]

\[
= DG \sqrt{(1 + 2\tau) T} \quad \text{for } \eta = \frac{D}{G \sqrt{(1 + 2\tau) T}}
\]
Delayed Feedback Analysis

1. Reduction to linear losses
2. Regret of OGD with delayed feedback $w_t$ is at most:
   - Regret of oracle OGD $w_t^*$ that observes all gradients
   - $\sum_{t=1}^{T} \left( \langle w_t, g_t \rangle - \langle w_t^*, g_t \rangle \right)$

$$= \sum_{t=1}^{T} \left( \langle w_1 - \eta \sum_{s \in [t-1]\setminus U_t} g_s, g_t \rangle - \langle w_1 - \eta \sum_{s \in [t-1]} g_s, g_t \rangle \right)$$

$$= \sum_{t=1}^{T} \langle \eta \sum_{s \in U_t} g_s, g_t \rangle$$

$$\leq \eta \sum_{t=1}^{T} \|g_t\| \sum_{s \in U_t} \|g_s\|$$

$$\text{Regret}_T(u) \leq \frac{1}{2\eta} \|w_1 - u\|^2 + \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|^2 + \eta \sum_{t=1}^{T} \|g_t\| \sum_{s \in U_t} \|g_s\|$$
Distributed Online Convex Optimization

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

Given connection graph $G$ between $N$ agents:

1. for $t = 1, 2, \ldots, T$ do
2. Nature activates agent $l_t \in \{1, \ldots, N\}$
3. Active agent $l_t$ predicts $w_t \in \mathcal{W}$
4. Nature reveals convex loss function $f_t : \mathcal{W} \rightarrow \mathbb{R}$ only to agent $l_t$
5. All agents can send a message to their neighbors in $G$
6. end for

Agents cooperate to minimize joint regret:

$$
\text{Regret}_T(u) = \sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(u)
$$
Distributed Learning Causes Delayed Feedback

Incurring the maximum delay:

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

  $$\text{Regret}_T(u) = O\left(D \sqrt{\text{diam}(\mathcal{G}) T}\right)$$
Distributed Learning Causes Delayed Feedback

Incurring the maximum delay:

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

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

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

\[
\text{Regret}_T(\mathbf{u}) = O\left(DG \sqrt{\text{diam}(\mathcal{G})T} \right)
\]

This is suboptimal:

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

\textbf{Much better:} Learn separately for each cluster:

\[
\text{Regret}_T(\mathbf{u}) = O\left(DG \sqrt{\text{diam}(\mathcal{F}_1)T} + DG \sqrt{\text{diam}(\mathcal{F}_2)T} \right)
\]

But optimal clustering depends on activations. How do we learn it?
Learning the Best Graph Partition

Given collection $Q$ of subgraphs of $G$, a $Q$-partition is a partition $\{F_1, \ldots, F_r\}$ of $G$ such that each $F_i \in Q$.

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

Given any $Q$, there exists an algorithm that guarantees

$$\sum_{j=1}^{r} \text{Regret}_{F_j}(u_j) = O\left(\sum_{j=1}^{r} \|u_j\|_G \left(\sqrt{\text{diam}(F_j) T_j \ln(1 + |Q| \text{diam}(F_j) \|u_j\|_T)}\right)\right)$$

for any $Q$-partition $\{F_1, \ldots, F_r\}$ and any $u_1, \ldots, u_r \in \mathcal{W}$.

$$\text{Regret}_{F_j}(u) = \sum_{t: l_t \in F_j} (f_t(w_t) - f_t(u))$$
Comparator-Adaptive Algorithms

Unbounded domain:

- Regret\( T(u) = O(DG\sqrt{T}) \) when comparator \( u \in \mathcal{W} \) with diameter of \( \mathcal{W} \) at most \( D \).
- What if we have no bound a priori on comparator norm \( \|u\| \), 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

\[
\text{Regret}_T(u) = O(\|u\|G\sqrt{T \log \left( \frac{T + \|u\|}{\epsilon} \right)} + \epsilon G) \quad \text{for all } u \in \mathbb{R}^d.
\]

- Essentially as good as bounded domain \( \mathcal{W} = \{w : \|w\| \leq \frac{1}{2}D\} \) for oracle choice \( D = 2\|u\| \).
Aggregating Multiple Online Methods

Aggregation:

- Given $K$ online learning algorithms with iterates $w_1^t, \ldots, w^K_t$
- Predict almost as well as the best one $k^*$:
  \[
  \text{Regret}_T(u) \leq \text{Regret}_{T}^{k^*}(u) + \text{overhead}
  \]
Aggregating Multiple Online Methods

Aggregation:
- Given $K$ online learning algorithms with iterates $w_t^1, \ldots, w_t^K$
- Predict almost as well as the best one $k^*$:
  \[
  \text{Regret}_T(u) \leq \text{Regret}_{k^*}^T(u) + \text{overhead}
  \]

Results: [Littlestone, Warmuth, 1994], [Vovk, 1998]: If $f_t(w_t^k) \in [a, b]$, then can achieve
  \[
  \text{overhead} = O((b - a)\sqrt{T \ln K})
  \]
Aggregating Multiple Online Methods

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

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

Results: [Littlestone, Warmuth, 1994], [Vovk, 1998]: If $f_t(w^k_t) \in [a, b]$, 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 $w_t = \sum_{k=1}^K w^k_t$ achieves

$$\text{overhead} = \sum_{k \neq k^*} \text{Regret}^k_T(0) = O(\epsilon KG) \quad \text{think: } \epsilon \propto 1/K$$
Aggregating Multiple Online Methods

Aggregation:
- Given \( K \) online learning algorithms with iterates \( w_1^t, \ldots, w^K_t \)
- Predict almost as well as the best one \( k^* \):

\[
\text{Regret}_T(u) \leq \text{Regret}^*_T(u) + \text{overhead}
\]

Results: [Littlestone, Warmuth, 1994], [Vovk, 1998]: If \( f_t(w^k_t) \in [a, b] \), 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 \( w_t = \sum_{k=1}^{K} w^k_t \) achieves

\[
\text{overhead} = \sum_{k \neq k^*} \text{Regret}^*_T(0) = O(\epsilon KG) \quad \text{think: } \epsilon \propto 1/K
\]

Proof:
\[
\sum_{t=1}^{T} \langle w_t, g_t \rangle - \langle u, g_t \rangle = \sum_{k=1}^{K} \sum_{t=1}^{T} \langle w^k_t, g_t \rangle - \sum_{t=1}^{T} \langle u, g_t \rangle
\]

\[
= \sum_{t=1}^{T} \left( \langle w^k_t, g_t \rangle - \langle u, g_t \rangle \right) + \sum_{k \neq k^*} \sum_{t=1}^{T} \left( \langle w^k_t, g_t \rangle - \langle 0, g_t \rangle \right)
\]
Learning the Graph Partition: Approach

Challenge:

▶ For each node \( i \) in the graph and cell \( F_j \in Q \) that contains \( i \), construct an algorithm \( w_t^{(i,j)} \) that can handle delays \( \tau = \text{diam}(F_j) \)

▶ Then \( i \) aggregates iterates \( w_t^{(i,j)} \) for all such \( j \)

▶ Problem: standard aggregation techniques with delays incur overhead that depends on maximum delay \( \max_j \text{diam}(F_j) \)

Our Solution:

▶ Make sure that \( w_t^{(i,j)} \) not only can handle delays, but are also comparator adaptive (new result)

▶ Then aggregation is possible using iterate addition, with overhead that depends on \( \text{diam}(F_j) \) for optimal \( F_j \).

▶ Project \( w_t \) onto bounded \( W \) using black-box reduction by \cite{Cutkosky, Orabona, 2018}
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 $w^{(i,j)}_t$ that can handle delays $\tau = \text{diam}(\mathcal{F}_j)$
- Then $i$ aggregates iterates $w^{(i,j)}_t$ for all such $j$
- Problem: standard aggregation techniques with delays incur overhead that depends on maximum delay $\max_j \text{diam}(\mathcal{F}_j)$

Our Solution:
- Make sure that $w^{(i,j)}_t$ not only can handle delays, but are also comparator adaptive (new result)
- Then aggregation is possible using iterate addition, with overhead that depends on $\text{diam}(\mathcal{F}_j)$ for optimal $\mathcal{F}_j$.
- Project $w_t$ 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