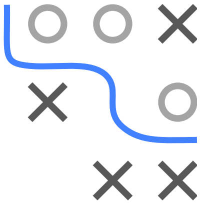
THE ONLINE LEARNER



- In the following, we will consider a first (online) learner for online learning problems. Note that a learner can be defined in a formal way.

THE ONLINE LEARNER

- In the following, we will consider a first (online) learner for online learning problems. Note that a learner can be defined in a formal way.
- Indeed, a learner within the basic online learning protocol, say `Algo`, is a function

$$A : \bigcup_{t=1}^{T} (\mathcal{Z} \times \mathcal{A})^t \to \mathcal{A}$$

that returns the current action based on (the loss and) the full history of information so far:

$$a_{t+1}^{\texttt{Algo}} = A(z_1, a_1^{\texttt{Algo}}, z_2, a_2^{\texttt{Algo}}, \ldots, z_t, a_t^{\texttt{Algo}};).$$

THE ONLINE LEARNER

- In the following, we will consider a first (online) learner for online learning problems. Note that a learner can be defined in a formal way.
- Indeed, a learner within the basic online learning protocol, say $\mathtt{Algo}$, is a function

$$A : \bigcup_{t=1}^{T} (\mathcal{Z} \times \mathcal{A})^t \to \mathcal{A}$$

  that returns the current action based on (the loss and) the full history of information so far:

$$a_{t+1}^{\mathtt{Algo}} = A(z_1, a_1^{\mathtt{Algo}}, z_2, a_2^{\mathtt{Algo}}, \ldots, z_t, a_t^{\mathtt{Algo}};).$$

- In the extended online learning scenario, where the environmental data consists of two parts, $z_t = (z_t^{(1)}, z_t^{(2)})$, and the first part is revealed before the action in t is performed, we have that

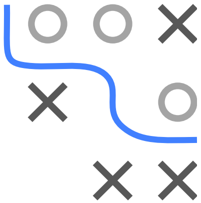
$$a_{t+1}^{\mathtt{Algo}} = A(z_1, a_1^{\mathtt{Algo}}, z_2, a_2^{\mathtt{Algo}}, \ldots, z_t, a_t^{\mathtt{Algo}}, z_{t+1}^{(1)};)$$

THE ONLINE LEARNER

- It will be desired that the online learner admits a *cheap update formula*, which is incremental, i.e., only a portion of the previous data is necessary to determine the next action.
- For instance, there exists a function $u : \mathcal{Z} \times \mathcal{A} \to \mathcal{A}$ such that

$$A(z_1, a_1^{\texttt{Algo}}, z_2, a_2^{\texttt{Algo}}, \ldots, z_t, a_t^{\texttt{Algo}};) = u(z_t, a_t^{\texttt{Algo}}).$$

FOLLOW THE LEADER ALGORITHM



- A simple algorithm to tackle online learning problems is the **Follow the leader** (FTL) algorithm.
- The algorithm takes as its action $a_t^{\mathtt{FTL}} \in \mathcal{A}$ in time step $t \geq 2$, the element which has the minimal cumulative loss so far over the previous $t-1$ time periods:

$$a_t^{\mathtt{FTL}} \in \operatorname*{arg\,min}_{a \in \mathcal{A}} \sum_{s=1}^{t-1} (a, z_s).$$

(Technical side note: if there are more than one minimum, then one of them is chosen. Moreover, $a_1^{\mathtt{FTL}}$ is arbitrary.)

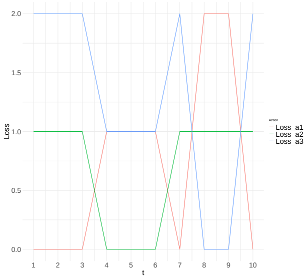
FOLLOW THE LEADER ALGORITHM

- A simple algorithm to tackle online learning problems is the **Follow the leader** (FTL) algorithm.
- The algorithm takes as its action $a_t^{\mathtt{FTL}} \in \mathcal{A}$ in time step $t \geq 2$, the element which has the minimal cumulative loss so far over the previous $t-1$ time periods:

$$a_t^{\mathtt{FTL}} \in \arg\min_{a \in \mathcal{A}} \sum_{s=1}^{t-1} (a, z_s).$$

(Technical side note: if there are more than one minimum, then one of them is chosen. Moreover, $a_1^{\mathtt{FTL}}$ is arbitrary.)

- *Interpretation:* The action $a_t^{\mathtt{FTL}}$ is the current "leader" of the actions in $\mathcal{A}$ in time step t, as it has the smallest cumulative loss (error) so far.



FOLLOW THE LEADER ALGORITHM

$$a_t^{\mathrm{FTL}} \in \underset{a \in \mathcal{A}}{\arg\min} \sum_{s=1}^{t-1} (a, z_s).$$

- Note that the action selection rule of FTL is natural and has much in common with the classical batch learning approaches based on empirical risk minimization.
- This results in a first issue regarding the computation time for the action, because the longer we run this algorithm, the slower it becomes (in general) due to the growth of the seen data.

FTL: A HELPFUL LEMMA

Lemma: Let $a_1^{\mathtt{FTL}}, a_2^{\mathtt{FTL}}, \ldots$ be the sequence of actions used by the FTL algorithm for the environmental data sequence $z_1, z_2, \ldots$.

FTL: A HELPFUL LEMMA

Lemma: Let $a_1^{\mathrm{FTL}}, a_2^{\mathrm{FTL}}, \ldots$ be the sequence of actions used by the FTL algorithm for the environmental data sequence $z_1, z_2, \ldots$.
Then, for all $\tilde{a} \in \mathcal{A}$ it holds that

$$\begin{aligned} R_T^{\mathrm{FTL}}(\tilde{a}) &= \sum\nolimits_{t=1}^{T} \left((a_t^{\mathrm{FTL}}, z_t) - (\tilde{a}, z_t)\right) \\ &\leq \sum\nolimits_{t=1}^{T} \left((a_t^{\mathrm{FTL}}, z_t) - (a_{t+1}^{\mathrm{FTL}}, z_t)\right) \\ &= \sum\nolimits_{t=1}^{T} (a_t^{\mathrm{FTL}}, z_t) - \sum\nolimits_{t=1}^{T} (a_{t+1}^{\mathrm{FTL}}, z_t). \end{aligned}$$

FTL: A HELPFUL LEMMA

Lemma: Let $a_1^{\mathrm{FTL}}, a_2^{\mathrm{FTL}}, \ldots$ be the sequence of actions used by the FTL algorithm for the environmental data sequence $z_1, z_2, \ldots$.
Then, for all $\tilde{a} \in \mathcal{A}$ it holds that

$$
\begin{aligned}
R_T^{\mathrm{FTL}}(\tilde{a}) &= \sum\nolimits_{t=1}^{T} \left((a_t^{\mathrm{FTL}}, z_t) - (\tilde{a}, z_t)\right) \\
&\leq \sum\nolimits_{t=1}^{T} \left((a_t^{\mathrm{FTL}}, z_t) - (a_{t+1}^{\mathrm{FTL}}, z_t)\right) \\
&= \sum\nolimits_{t=1}^{T} (a_t^{\mathrm{FTL}}, z_t) - \sum\nolimits_{t=1}^{T} (a_{t+1}^{\mathrm{FTL}}, z_t).
\end{aligned}
$$

In particular,

$$
R_T^{\mathrm{FTL}} \leq \sum\nolimits_{t=1}^{T} (a_t^{\mathrm{FTL}}, z_t) - \sum\nolimits_{t=1}^{T} (a_{t+1}^{\mathrm{FTL}}, z_t)
$$

FTL: A HELPFUL LEMMA

Lemma: Let $a_1^{\text{FTL}}, a_2^{\text{FTL}}, \dots$ be the sequence of actions used by the FTL algorithm for the environmental data sequence $z_1, z_2, \dots$.
Then, for all $\tilde{a} \in \mathcal{A}$ it holds that

$$\begin{aligned} R_T^{\text{FTL}}(\tilde{a}) &= \sum\nolimits_{t=1}^{T} \left((a_t^{\text{FTL}}, z_t) - (\tilde{a}, z_t)\right) \\ &\leq \sum\nolimits_{t=1}^{T} \left((a_t^{\text{FTL}}, z_t) - (a_{t+1}^{\text{FTL}}, z_t)\right) \\ &= \sum\nolimits_{t=1}^{T} (a_t^{\text{FTL}}, z_t) - \sum\nolimits_{t=1}^{T} (a_{t+1}^{\text{FTL}}, z_t). \end{aligned}$$

In particular,

$$R_T^{\text{FTL}} \leq \sum\nolimits_{t=1}^{T} (a_t^{\text{FTL}}, z_t) - \sum\nolimits_{t=1}^{T} (a_{t+1}^{\text{FTL}}, z_t)$$

Interpretation: the regret of the FTL algorithm is bounded by the difference of cumulated losses of itself compared to its one-step lookahead cheater version.

FTL: A HELPFUL LEMMA

Proof: In the following, we denote $a_1^{\mathrm{FTL}}, a_2^{\mathrm{FTL}}, \ldots$ simply by $a_1, a_2, \ldots$

FTL: A HELPFUL LEMMA

Proof: In the following, we denote $a_1^{\mathrm{FTL}}, a_2^{\mathrm{FTL}}, \ldots$ simply by $a_1, a_2, \ldots$
First, note that the assertion can be restated as follows

$$R_T^{\mathrm{FTL}}(\tilde{a}) = \sum_{t=1}^{T} ((a_t, z_t) - (\tilde{a}, z_t)) \leq \sum_{t=1}^{T} ((a_t, z_t) - (a_{t+1}, z_t))$$

$$\Leftrightarrow \sum_{t=1}^{T} (a_{t+1}, z_t) \leq \sum_{t=1}^{T} (\tilde{a}, z_t).$$

FTL: A HELPFUL LEMMA

Proof: In the following, we denote $a_1^{\mathrm{FTL}}, a_2^{\mathrm{FTL}}, \ldots$ simply by $a_1, a_2, \ldots$
First, note that the assertion can be restated as follows

$$R_T^{\mathrm{FTL}}(\tilde{a}) = \sum_{t=1}^{T} ((a_t, z_t) - (\tilde{a}, z_t)) \leq \sum_{t=1}^{T} ((a_t, z_t) - (a_{t+1}, z_t))$$
$$\Leftrightarrow \sum_{t=1}^{T} (a_{t+1}, z_t) \leq \sum_{t=1}^{T} (\tilde{a}, z_t).$$

Hence, we will verify the inequality $\sum_{t=1}^{T} (a_{t+1}, z_t) \leq \sum_{t=1}^{T} (\tilde{a}, z_t)$, which implies the assertion.

$\rightsquigarrow$ This will be done by induction over T.

FTL: A HELPFUL LEMMA

Reminder: $$a_t^{\mathrm{FTL}} \in \arg\min_{a \in \mathcal{A}} \sum_{s=1}^{t-1} (a, z_s).$$

Initial step: $T = 1$. It holds that

$$\sum_{t=1}^{T} (a_{t+1}, z_t) = (a_2, z_1) = \left(\arg\min_{a \in \mathcal{A}} (a, z_1), z_1 \right)$$

$$= \min_{a \in \mathcal{A}} (a, z_1) \leq (\tilde{a}, z_1) \quad \left(= \sum\nolimits_{t=1}^{T} (\tilde{a}, z_t) \right)$$

for all $\tilde{a} \in \mathcal{A}$.

FTL: A HELPFUL LEMMA

Reminder: $$a_t^{\mathrm{FTL}} \in \arg\min_{a \in \mathcal{A}} \sum_{s=1}^{t-1} (a, z_s).$$

Initial step: $T = 1$. It holds that

$$\sum_{t=1}^{T} (a_{t+1}, z_t) = (a_2, z_1) = \left(\arg\min_{a \in \mathcal{A}} (a, z_1), z_1 \right)$$

$$= \min_{a \in \mathcal{A}} (a, z_1) \leq (\tilde{a}, z_1) \quad \left(= \sum\nolimits_{t=1}^{T} (\tilde{a}, z_t) \right)$$

for all $\tilde{a} \in \mathcal{A}$.

Induction Step: $T - 1 \rightarrow T$. Assume that for any $\tilde{a} \in \mathcal{A}$ it holds that

$$\sum\nolimits_{t=1}^{T-1} (a_{t+1}, z_t) \leq \sum\nolimits_{t=1}^{T-1} (\tilde{a}, z_t).$$

Then, the following holds as well (adding (a_{T+1}, z_T) on both sides)

$$\sum\nolimits_{t=1}^{T} (a_{t+1}, z_t) \leq (a_{T+1}, z_T) + \sum\nolimits_{t=1}^{T-1} (\tilde{a}, z_t), \quad \forall \tilde{a} \in \mathcal{A}.$$

FTL: A HELPFUL LEMMA

Reminder (1): $\sum_{t=1}^{T} (a_{t+1}, z_t) \leq (a_{T+1}, z_T) + \sum_{t=1}^{T-1} (\tilde{a}, z_t).$

Reminder (2): $a_t^{\mathrm{FTL}} \in \arg\min_{a \in \mathcal{A}} \sum_{s=1}^{t-1} (a, z_s).$

FTL: A HELPFUL LEMMA

Reminder (1): $\sum_{t=1}^{T} (a_{t+1}, z_t) \leq (a_{T+1}, z_T) + \sum_{t=1}^{T-1} (\tilde{a}, z_t).$

Reminder (2): $a_t^{\mathrm{FTL}} \in \arg\min_{a \in \mathcal{A}} \sum_{s=1}^{t-1} (a, z_s).$

Using (1) with $\tilde{a} = a_{T+1}$ yields

$$
\begin{aligned}
\sum_{t=1}^{T} (a_{t+1}, z_t) &\leq \sum_{t=1}^{T} (a_{T+1}, z_t) = \sum_{t=1}^{T} \Big(\operatorname*{arg\,min}_{a \in \mathcal{A}} \sum_{t=1}^{T} (a, z_t), z_t \Big) \\
&= \min_{a \in \mathcal{A}} \sum_{t=1}^{T} (a, z_t) \leq \sum_{t=1}^{T} (\tilde{a}, z_t)
\end{aligned}
$$

for all $\tilde{a} \in \mathcal{A}$. $\square$

FTL FOR OQO PROBLEMS

- One popular instantiation of the online learning problem is the problem of *online quadratic optimization* (OQO).
- In its most general form, the loss function is thereby defined as

$$(a_t, z_t) = \frac{1}{2} \left|\left| a_t - z_t \right|\right|_2^2 ,$$

where $\mathcal{A}, \mathcal{Z} \subset \mathbb{R}^d$.

FTL FOR OQO PROBLEMS

- One popular instantiation of the online learning problem is the problem of *online quadratic optimization* (OQO).
- In its most general form, the loss function is thereby defined as

$$(a_t, z_t) = \frac{1}{2}\,||a_t - z_t||_2^2\,,$$

where $\mathcal{A}, \mathcal{Z} \subset \mathbb{R}^d$.

- **Proposition:** Using FTL on any online quadratic optimization problem with $\mathcal{A} = \mathbb{R}^d$ and $V = \sup_{z \in \mathcal{Z}} ||z||_2$, leads to a regret of

$$R_T^{\mathtt{FTL}} \leq 4V^2\,(\log(T) + 1).$$

FTL FOR OQO PROBLEMS

- One popular instantiation of the online learning problem is the problem of *online quadratic optimization* (OQO).
- In its most general form, the loss function is thereby defined as

$$(a_t, z_t) = \frac{1}{2} \, ||a_t - z_t||_2^2 \, ,$$

  where $\mathcal{A}, \mathcal{Z} \subset \mathbb{R}^d$.
- **Proposition:** Using FTL on any online quadratic optimization problem with $\mathcal{A} = \mathbb{R}^d$ and $V = \sup_{z \in \mathcal{Z}} ||z||_2$, leads to a regret of

$$R_T^{\mathtt{FTL}} \leq 4V^2 \left(\log(T) + 1\right).$$

- This result is satisfactory for three reasons:
  1. The regret is definitely sublinear, that is, $R_T^{\mathtt{FTL}} = o(T)$.
  2. We just have a mild constraint on the online quadratic optimization problem, namely that $||z||_2 \leq V$ holds for any possible environmental data instance $z \in \mathcal{Z}$.
  3. The action $a_t^{\mathtt{FTL}}$ is simply the empirical average of the environmental data seen so far: $a_t^{\mathtt{FTL}} = \frac{1}{t-1} \sum_{s=1}^{t-1} z_s$.