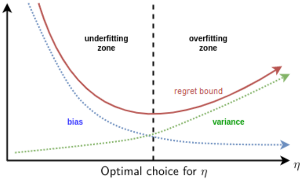
Advanced Machine Learning

Simple Online Learning Algorithms



Learning goals

- Get to know the FTL algorithm
- See it’s flaws and understand the root cause
- Get to know the FTRL algorithm as a stable alternative

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 L and) the full history of information so far:

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

- 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

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

- 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

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

FOLLOW THE LEADER ALGORITHM

- A simple algorithm to tackle online learning problems is the **Follow the leader** (FTL) algorithm.
- Suppose that the online learning problem consists of an action space $\mathcal{A} \subset \mathbb{R}^{d_1}$, an environmental space $\mathcal{Z} \subset \mathbb{R}^{d_2}$ and some loss function $L : \mathcal{A} \times \mathcal{Z} \to \mathbb{R}$ with $d_1, d_2 \in \mathbb{N}$.
- The algorithm takes as its action $a_t^{\mathrm{FTL}} \in \mathcal{A}$ in time step $t \geq 2$, the element which has the minimal cumulative loss so far over the previous $t - 1$ time periods:

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

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

- *Interpretation:* The action a_t^{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 simple algorithm to tackle online learning problems is the **Follow the leader** (FTL) algorithm.
- Suppose that the online learning problem consists of an action space $\mathcal{A} \subset \mathbb{R}^{d_1}$, an environmental space $\mathcal{Z} \subset \mathbb{R}^{d_2}$ and some loss function $L : \mathcal{A} \times \mathcal{Z} \to \mathbb{R}$ with $d_1, d_2 \in \mathbb{N}$.
- The algorithm takes as its action $a_t^{\mathrm{FTL}} \in \mathcal{A}$ in time step $t \geq 2$, the element which has the minimal cumulative loss so far over the previous $t-1$ time periods:

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

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

- *Interpretation:* The action a_t^{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.
- 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 $a_1^{\mathrm{FTL}}, a_2^{\mathrm{FTL}}, \ldots$ 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

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

In particular,

$$R_T^{\mathrm{FTL}} \leq \sum_{t=1}^{T} \left(L(a_t^{\mathrm{FTL}}, z_t) - L(a_{t+1}^{\mathrm{FTL}}, z_t)) \right).$$

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 $a_1^{\mathtt{FTL}}, a_2^{\mathtt{FTL}}, \ldots$ simply by $a_1, a_2, \ldots$
First, note that the assertion can be restated as follows

$$\sum_{t=1}^{T} \left(L(a_t, z_t) - L(\tilde{a}, z_t)\right) \leq \sum_{t=1}^{T} \left(L(a_t, z_t) - L(a_{t+1}, z_t)\right)$$

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

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

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

FTL: A HELPFUL LEMMA

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

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

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

FTL: A HELPFUL LEMMA

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

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

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

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

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

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

FTL: A HELPFUL LEMMA

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

FTL: A HELPFUL LEMMA

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

Using the latter inequality with $\tilde{a} = a_{T+1}$ yields

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

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

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

where $\mathcal{A}$ and $\mathcal{Z}$ have the same dimensionality.

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

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

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

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

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

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

  where $\mathcal{A}$ and $\mathcal{Z}$ have the same dimensionality.
- **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

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

- This result is satisfactory for three reasons:
  - The regret is definitely sublinear, that is, $R_T^{\mathrm{FTL}} = o(T)$.
  - 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 $z \in \mathcal{Z}$.
  - The action a_t^{FTL} is simply the empirical average of the environmental data seen so far: $a_t^{\mathrm{FTL}} = \frac{1}{t-1} \sum_{s=1}^{t-1} z_s$.

FTL FOR ONLINE LINEAR OPTIMIZATION

- Another popular instantiation of the online learning problem is the online linear optimization problem, which is characterized by a linear loss function $L(a, z) = a^\top z$.
- Let $\mathcal{A} = [-1, 1]$ and suppose that $z_t = \begin{cases} -\frac{1}{2}, & t = 1, \\ 1, & t \text{ is even}, \\ -1, & t \text{ is odd}. \end{cases}$
- No matter how we choose the first action $a_1^{\mathtt{FTL}}$, it will hold that FTL has a cumulative loss greater than (or equal) $T - 3/2$, while the best action in hindsight has a cumulative loss of $-1/2$.
- Thus, FTL's cumulative regret is at least $T - 1$, which is linearly growing in T.

FTL FOR ONLINE LINEAR OPTIMIZATION

- Indeed, note that

$$
\begin{aligned}
a_{t+1}^{\mathrm{FTL}} = \arg\min_{a\in\mathcal{A}} \sum_{s=1}^{t} L(a, z_s) &= \arg\min_{a\in[-1,1]} a \sum_{s=1}^{t} z_s \\
&= \begin{cases} -1, & \text{if } \sum_{s=1}^{t} z_s > 0, \\ 1, & \text{if } \sum_{s=1}^{t} z_s < 0, \\ \text{arbitrary}, & \text{if } \sum_{s=1}^{t} z_s = 0. \end{cases}
\end{aligned}
$$

t	a_t^{FTL}	z_t	$L(a_t^{\mathrm{FTL}}, z_t)$	$\sum_{s=1}^{t} L(a_s^{\mathrm{FTL}}, z_s)$	$\sum_{s=1}^{t} z_s$
1	1	$-1/2$	$-1/2$	$-1/2$	$-1/2$
2	1	1	1	$1 -1/2$	1/2
3	-1	-1	1	$2 -1/2$	$-1/2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
T	$(-1)^T$	$(-1)^T$	1	$T-1-1/2$	$(-1/2)^T$

- The best action has cumulative loss

$$
\inf_{a\in\mathcal{A}} \sum_{s=1}^{T} L(a, z_s) = \inf_{a\in[-1,1]} a \underbrace{\sum_{s=1}^{T} z_s}_{=(-1/2)^T} = -1/2.
$$

FTL FOR ONLINE LINEAR OPTIMIZATION

- Thus, we see: FTL can fail for online linear optimization problems, although it is well suited for online quadratic optimization problems!
- The reason is that the action selection of FTL is not stable enough (caused by the loss function), which is fine for the latter problem, but problematic for the former.
- One has to note that the online linear optimization problem example above, where FTL fails, is in fact an adversarial learning setting: The environmental data is generated in such a way that the FTL learner is fooled in each time step.

FOLLOW THE REGULARIZED LEADER

- To overcome the shortcomings of the FTL algorithm, one can incorporate a regularization function $\psi : \mathcal{A} \to \mathbb{R}_+$ into the action choice of FTL, which leads to more stability.
- To be more precise, let for $t \geq 1$

$$a_t^{\mathrm{FTRL}} \in \arg\min_{a \in \mathcal{A}} \left(\psi(a) + \sum_{s=1}^{t-1} L(a, z_s) \right),$$

(Technical side note: if there are more than one minimum, then one of them is chosen.)

then the algorithm choosing a_t^{FTRL} in time step t is called the **Follow the regularized leader** (FTRL) algorithm.

FOLLOW THE REGULARIZED LEADER

- To overcome the shortcomings of the FTL algorithm, one can incorporate a regularization function $\psi : \mathcal{A} \to \mathbb{R}_+$ into the action choice of FTL, which leads to more stability.
- To be more precise, let for $t \geq 1$

$$a_t^{\mathrm{FTRL}} \in \arg\min_{a \in \mathcal{A}} \left(\psi(a) + \sum_{s=1}^{t-1} L(a, z_s) \right),$$

(Technical side note: if there are more than one minimum, then one of them is chosen.)

then the algorithm choosing a_t^{FTRL} in time step t is called the **Follow the regularized leader** (FTRL) algorithm.

- *Interpretation:* The algorithm predicts a_t as the element in $\mathcal{A}$, which minimizes the regularization function plus the cumulative loss so far over the previous $t - 1$ time periods.

FOLLOW THE REGULARIZED LEADER

- To overcome the shortcomings of the FTL algorithm, one can incorporate a regularization function $\psi : \mathcal{A} \to \mathbb{R}_+$ into the action choice of FTL, which leads to more stability.
- To be more precise, let for $t \geq 1$

$$a_t^{\mathrm{FTRL}} \in \arg\min_{a \in \mathcal{A}} \left(\psi(a) + \sum_{s=1}^{t-1} L(a, z_s) \right),$$

(Technical side note: if there are more than one minimum, then one of them is chosen.)

then the algorithm choosing a_t^{FTRL} in time step t is called the **Follow the regularized leader** (FTRL) algorithm.

- *Interpretation:* The algorithm predicts a_t as the element in $\mathcal{A}$, which minimizes the regularization function plus the cumulative loss so far over the previous $t-1$ time periods.
- Obviously, the behavior of the FTRL algorithm is depending heavily on the choice of the regularization function ψ. If $\psi \equiv 0$, then FTRL equals FTL.

REGULARIZATION IN ONLINE LEARNING VS. BATCH LEARNING

- Note that in the batch learning scenario, i.e., where training phase and testing phase are decoupled and training data is available before the testing phase, the learner seeks to optimize an objective function which is the sum of the training loss and a regularization function:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^p} \sum_{i=1}^{n} L(y^{(i)}, \boldsymbol{\theta}) + \lambda \, \psi(\boldsymbol{\theta}),$$

where $\lambda \geq 0$ is some regularization parameter.

REGULARIZATION IN ONLINE LEARNING VS. BATCH LEARNING

- Note that in the batch learning scenario, i.e., where training phase and testing phase are decoupled and training data is available before the testing phase, the learner seeks to optimize an objective function which is the sum of the training loss and a regularization function:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^p} \sum_{i=1}^{n} L(y^{(i)}, \boldsymbol{\theta}) + \lambda\, \psi(\boldsymbol{\theta}),$$

  where $\lambda \geq 0$ is some regularization parameter.

- Here, the regularization function is part of the whole objective function, which the learner seeks to minimize.

REGULARIZATION IN ONLINE LEARNING VS. BATCH LEARNING

- Note that in the batch learning scenario, i.e., where training phase and testing phase are decoupled and training data is available before the testing phase, the learner seeks to optimize an objective function which is the sum of the training loss and a regularization function:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^p} \sum_{i=1}^{n} L(y^{(i)}, \boldsymbol{\theta}) + \lambda \, \psi(\boldsymbol{\theta}),$$

  where $\lambda \geq 0$ is some regularization parameter.

- Here, the regularization function is part of the whole objective function, which the learner seeks to minimize.
- However, in the online learning scenario the regularization function does (usually) not appear in the regret the learner seeks to minimize, but the regularization function is only part of the action/decision rule at each time step.

REGRET ANALYSIS OF FTRL: A HELPFUL LEMMA

- **Lemma:** Let $a_1^{\mathtt{FTRL}}, a_2^{\mathtt{FTRL}}, \ldots$ be the sequence of actions coming used by the FTRL algorithm for the environmental data sequence $z_1, z_2, \ldots$. Then, for all $\tilde{a} \in \mathcal{A}$ we have

$$
\begin{aligned}
R_T^{\mathtt{FTRL}}(\tilde{a}) &= \sum_{t=1}^{T} \left(L(a_t^{\mathtt{FTRL}}, z_t) - L(\tilde{a}, z_t) \right) \\
&\leq \psi(\tilde{a}) - \psi(a_1^{\mathtt{FTRL}}) + \sum_{t=1}^{T} \left(L(a_t^{\mathtt{FTRL}}, z_t) - L(a_{t+1}^{\mathtt{FTRL}}, z_t) \right).
\end{aligned}
$$

- *Interpretation*: the regret of the FTRL algorithm is bounded by the difference of cumulated losses of itself compared to its one-step lookahead cheater version and an additional regularization difference term.

$\Rightarrow$ We have seen an analogous result for FTL!

(The proof is similar.)

FTRL FOR ONLINE LINEAR OPTIMIZATION

- In the following, we analyze the FTRL algorithm for the linear loss $L(a, z) = a^\top z$ for online linear optimization (OLO) problems.
- For this purpose, the squared L2-norm regularization will be used, which is given by

$$\psi(a) = \frac{1}{2\eta} \, ||a||_2^2 = \frac{a^\top a}{2\eta},$$

  where η is some positive scalar, the *regularization magnitude.*

FTRL FOR ONLINE LINEAR OPTIMIZATION

- In the following, we analyze the FTRL algorithm for the linear loss $L(a, z) = a^\top z$ for online linear optimization (OLO) problems.
- For this purpose, the squared L2-norm regularization will be used, which is given by

$$\psi(a) = \frac{1}{2\eta} \, ||a||_2^2 = \frac{a^\top a}{2\eta},$$

  where η is some positive scalar, the *regularization magnitude.*
- It is straightforward to compute that if $\mathcal{A} = \mathbb{R}^d$, then

$$a_t^{\mathtt{FTRL}} = -\eta \sum_{s=1}^{t-1} z_s.$$

  Hence, in this case we have for the FTRL algorithm the following update rule

$$a_{t+1}^{\mathtt{FTRL}} = a_t^{\mathtt{FTRL}} - \eta \, z_t, \qquad t = 1, \ldots, T-1.$$

FTRL FOR ONLINE LINEAR OPTIMIZATION

- In the following, we analyze the FTRL algorithm for the linear loss $L(a, z) = a^\top z$ for online linear optimization (OLO) problems.
- For this purpose, the squared L2-norm regularization will be used, which is given by

$$\psi(a) = \frac{1}{2\eta} \, ||a||_2^2 = \frac{a^\top a}{2\eta},$$

where η is some positive scalar, the *regularization magnitude.*

- It is straightforward to compute that if $\mathcal{A} = \mathbb{R}^d$, then

$$a_t^{\mathrm{FTRL}} = -\eta \sum_{s=1}^{t-1} z_s.$$

Hence, in this case we have for the FTRL algorithm the following update rule

$$a_{t+1}^{\mathrm{FTRL}} = a_t^{\mathrm{FTRL}} - \eta \, z_t, \qquad t = 1, \dots, T-1.$$

Interpretation: $-z_t$ is the *direction* in which the update of a_t^{FTRL} to a_{t+1}^{FTRL} is conducted with *step size* η in order to reduce the loss.

FTRL FOR OLO: THEORETICAL GUARANTEES

- **Proposition:** Using the FTRL algorithm with the squared L2-norm regularization on any online linear optimization (OLO) problem with $\mathcal{A} \subset \mathbb{R}^d$ leads to a regret of FTRL with respect to any action $\tilde{a} \in \mathcal{A}$ of

$$R_T^{FTRL}(\tilde{a}) \leq \frac{1}{2\eta} ||\tilde{a}||_2^2 + \eta \sum_{t=1}^{T} ||z_t||_2^2 \,.$$

FTRL FOR OLO: THEORETICAL GUARANTEES

- **Proposition:** Using the FTRL algorithm with the squared L2-norm regularization on any online linear optimization (OLO) problem with $\mathcal{A} \subset \mathbb{R}^d$ leads to a regret of FTRL with respect to any action $\tilde{a} \in \mathcal{A}$ of

$$R_T^{FTRL}(\tilde{a}) \leq \frac{1}{2\eta} ||\tilde{a}||_2^2 + \eta \sum_{t=1}^{T} ||z_t||_2^2 .$$

- We will show the result only for the case $\mathcal{A} = \mathbb{R}^d$.
- For the more general case, where $\mathcal{A}$ is a strict subset of $\mathbb{R}^d$, we need a slight modification of the update formula above:

$$a_t^{\mathrm{FTRL}} = \Pi_{\mathcal{A}}\big(-\eta \sum\nolimits_{i=1}^{t-1} z_i \big) = \arg\min_{a \in \mathcal{A}} \left|\left| a - \eta \sum\nolimits_{i=1}^{t-1} z_i \right|\right|_2^2 \Big\}.$$

In words, the action of the FTRL algorithm has to be projected onto the set $\mathcal{A}$. Here, $\Pi_{\mathcal{A}} : \mathbb{R}^d \to \mathcal{A}$ is the projection onto $\mathcal{A}$.

(The proof is essentially the same, except that the Cauchy-Schwarz inequality is used in between.)

FTRL FOR OLO: THEORETICAL GUARANTEES

- **Proof:**
  - For sake of brevity, we write $a_1, a_2, \dots$ for $a_1^{\mathtt{FTRL}}, a_2^{\mathtt{FTRL}}, \dots$
  - With this,

$$R_T^{FTRL}(\tilde{a}) \leq \psi(\tilde{a}) - \psi(a_1) + \sum_{t=1}^{T} (L(a_t, z_t) - L(a_{t+1}, z_t))$$

(Lemma one-step lookahead cheater)

$$\leq \frac{1}{2\eta} ||\tilde{a}||_2^2 + \sum_{t=1}^{T} (a_t^\top z_t - a_{t+1}^\top z_t)$$

($\psi(a_1) \geq 0$ and definition of ψ)

$$= \frac{1}{2\eta} ||\tilde{a}||_2^2 + \sum_{t=1}^{T} (a_t^\top - a_{t+1}^\top) z_t \qquad \text{(distributivity)}$$

$$= \frac{1}{2\eta} ||\tilde{a}||_2^2 + \eta \sum_{t=1}^{T} ||z_t||_2^2 .$$

(Update formula $a_{t+1} = a_t - \eta z_t$)

$\square$

FTRL FOR OLO: THEORETICAL GUARANTEES

- Interpretation of the terms in the proposition, i.e., of

$$R_T^{FTRL}(\tilde{a}) \leq \frac{1}{2\eta} ||\tilde{a}||_2^2 + \eta \sum_{t=1}^{T} ||z_t||_2^2 :$$

FTRL FOR OLO: THEORETICAL GUARANTEES

- Interpretation of the terms in the proposition, i.e., of

$$R_T^{FTRL}(\tilde{a}) \leq \frac{1}{2\eta}||\tilde{a}||_2^2 + \eta \sum_{t=1}^{T} ||z_t||_2^2 :$$

  - $||\tilde{a}||_2^2$ represents a *bias term:* The regret upper bound of FTRL is always biased by the term $||\tilde{a}||_2^2$. The impact of the bias term can be reduced by a higher regularization magnitude, i.e., a higher choice of η.

FTRL FOR OLO: THEORETICAL GUARANTEES

- Interpretation of the terms in the proposition, i.e., of

$$R_T^{FTRL}(\tilde{a}) \leq \frac{1}{2\eta}||\tilde{a}||_2^2 + \eta \sum_{t=1}^{T} ||z_t||_2^2 :$$

  - $||\tilde{a}||_2^2$ represents a *bias term:* The regret upper bound of FTRL is always biased by the term $||\tilde{a}||_2^2$. The impact of the bias term can be reduced by a higher regularization magnitude, i.e., a higher choice of η.
  - $\sum_{t=1}^{T} ||z_t||_2^2$ represents a *"variance" term*: The more the environment data z_t varies, the larger this term. Hence, for a high variance a smaller regularization magnitude is needed, i.e., a smaller choice of η.

FTRL FOR OLO: THEORETICAL GUARANTEES

- Interpretation of the terms in the proposition, i.e., of

$$R_T^{FTRL}(\tilde{a}) \leq \frac{1}{2\eta}||\tilde{a}||_2^2 + \eta \sum_{t=1}^{T} ||z_t||_2^2 :$$

  - $||\tilde{a}||_2^2$ represents a *bias term:* The regret upper bound of FTRL is always biased by the term $||\tilde{a}||_2^2$. The impact of the bias term can be reduced by a higher regularization magnitude, i.e., a higher choice of η.
  - $\sum_{t=1}^{T} ||z_t||_2^2$ represents a *"variance" term*: The more the environment data z_t varies, the larger this term. Hence, for a high variance a smaller regularization magnitude is needed, i.e., a smaller choice of η.

- Thus, we have a trade-off for the optimal choice of η : Making η large, leads to a smaller bias but at the expanse of a higher variance and making η small leads to a smaller variance at the expanse of a higher bias.

FTRL FOR OLO: THEORETICAL GUARANTEES

- Interpretation of the terms in the proposition, i.e., of

$$R_T^{FTRL}(\tilde{a}) \leq \frac{1}{2\eta}||\tilde{a}||_2^2 + \eta\sum_{t=1}^{T}||z_t||_2^2 :$$

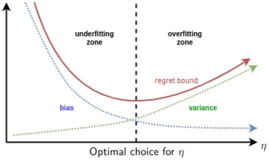
  - $||\tilde{a}||_2^2$ represents a *bias term:* The regret upper bound of FTRL is always biased by the term $||\tilde{a}||_2^2$. The impact of the bias term can be reduced by a higher regularization magnitude, i.e., a higher choice of η.
  - $\sum_{t=1}^{T}||z_t||_2^2$ represents a *"variance" term*: The more the environment data z_t varies, the larger this term. Hence, for a high variance a smaller regularization magnitude is needed, i.e., a smaller choice of η.

- Thus, we have a trade-off for the optimal choice of η : Making η large, leads to a smaller bias but at the expanse of a higher variance and making η small leads to a smaller variance at the expanse of a higher bias.

$\Rightarrow$ With the right choice of η, we can prevent the instability of FTRL for an online linear optimization (OLO) problem.

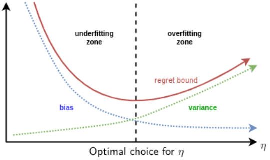
FTRL FOR OLO: THEORETICAL GUARANTEES

- Under certain assumptions we can balance the trade-off (see the picture) induced by the bias and the variance by choosing η appropriately.



FTRL FOR OLO: THEORETICAL GUARANTEES

- Under certain assumptions we can balance the trade-off (see the picture) induced by the bias and the variance by choosing η appropriately.



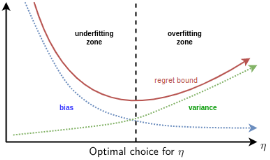
- **Corollary:** Suppose we use the FTRL algorithm with the squared L2-norm regularization on an online linear optimization problem with $\mathcal{A} \subset \mathbb{R}^d$ such that
  - $\sup_{\tilde{a} \in \mathcal{A}} ||\tilde{a}||_2 \leq B$ for some finite constant $B > 0$,
  - $\sup_{z \in \mathcal{Z}} ||z||_2 \leq V$ for some finite constant $V > 0$.

  Then, by choosing the step size η for FTRL as $\eta = \frac{B}{V\sqrt{2T}}$ it holds that

$$R_T^{FTRL} \leq BV\sqrt{2T}.$$

FTRL FOR OLO: THEORETICAL GUARANTEES

- Under certain assumptions we can balance the trade-off (see the picture) induced by the bias and the variance by choosing η appropriately.



- **Corollary:** Suppose we use the FTRL algorithm with the squared L2-norm regularization on an online linear optimization problem with $\mathcal{A} \subset \mathbb{R}^d$ such that
  - $\sup_{\tilde{a} \in \mathcal{A}} ||\tilde{a}||_2 \leq B$ for some finite constant $B > 0$,
  - $\sup_{z \in \mathcal{Z}} ||z||_2 \leq V$ for some finite constant $V > 0$.

  Then, by choosing the step size η for FTRL as $\eta = \frac{B}{V\sqrt{2T}}$ it holds that

$$R_T^{FTRL} \leq BV\sqrt{2T}.$$

- Note that the (optimal) parameter η depends on the time horizon T, which is oftentimes not known in advance. However, there are some tricks (i.e., the *doubling trick*), which can help in such cases.

FTRL FOR OLO: THEORETICAL GUARANTEES

- **Proof:**
  - By the latter proposition and the assumptions

$$
\begin{aligned}
R_T^{FTRL}(\tilde{a}) &\leq \frac{1}{2\eta} ||\tilde{a}||_2^2 + \eta \sum_{t=1}^{T} ||z_t||_2^2 \\
&\leq \frac{B^2}{2\eta} + \eta\, T\, V^2.
\end{aligned}
$$

FTRL FOR OLO: THEORETICAL GUARANTEES

- **Proof:**
  - By the latter proposition and the assumptions

$$
\begin{aligned}
R_T^{FTRL}(\tilde{a}) &\leq \frac{1}{2\eta}\,||\tilde{a}||_2^2 && + \eta \sum_{t=1}^{T} ||z_t||_2^2 \\
&\leq \frac{B^2}{2\eta} && + \eta\, T\, V^2.
\end{aligned}
$$

  - The right-hand side of the latter display is independent of $\tilde{a}$, so that

$$R_T^{FTRL} \leq \frac{B^2}{2\eta} + \eta\, T\, V^2.$$

FTRL FOR OLO: THEORETICAL GUARANTEES

- **Proof:**
  - By the latter proposition and the assumptions

$$
\begin{aligned}
R_T^{FTRL}(\tilde{a}) &\leq \frac{1}{2\eta}||\tilde{a}||_2^2 && + \eta \sum_{t=1}^{T} ||z_t||_2^2 \\
&\leq \frac{B^2}{2\eta} && + \eta\, T\, V^2.
\end{aligned}
$$

  - The right-hand side of the latter display is independent of $\tilde{a}$, so that

$$
R_T^{FTRL} \leq \frac{B^2}{2\eta} + \eta\, T\, V^2.
$$

  - Now, the right-hand side of the latter display is a function of the form $f(\eta) = a/\eta + b\eta$ for some suitable $a, b > 0$.
  - Minimizing f with respect to η results in the minimizer $\eta^* = \frac{B}{V\sqrt{2\,T}}$.
  - Plugging this minimizer into the latter display leads to the asserted inequality. □

DESIRED RESULTS

- With the FTRL algorithm we can cope with
  - online quadratic optimization (OQO) problems by using no regularity ($\psi \equiv 0$). In this case, we have satisfactory regret guarantees and also a quick update rule for a_{t+1}^{FTRL} (It is just the empirical average over all data points seen till t),

DESIRED RESULTS

- With the FTRL algorithm we can cope with
  - online quadratic optimization (OQO) problems by using no regularity ($\psi \equiv 0$). In this case, we have satisfactory regret guarantees and also a quick update rule for a_{t+1}^{FTRL} (It is just the empirical average over all data points seen till t),
  - online linear optimization (OLO) problems by using a suitable regularization function. In this case, we have quick update formulas and satisfactory regret guarantees as well.

DESIRED RESULTS

- With the FTRL algorithm we can cope with
  - online quadratic optimization (OQO) problems by using no regularity ($\psi \equiv 0$). In this case, we have satisfactory regret guarantees and also a quick update rule for a_{t+1}^{FTRL} (It is just the empirical average over all data points seen till t),
  - online linear optimization (OLO) problems by using a suitable regularization function. In this case, we have quick update formulas and satisfactory regret guarantees as well.

$\Rightarrow$ But what about other online learning problems or rather other loss functions?

- What we wish to have is an approach such that we can achieve for a large class of loss functions L the advantages of FTRL for OLO and OCO problems:

  (a) reasonable regret upper bounds;
  (b) a quick update formula.