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 $t \geq 1$

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

(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 $t \geq 1$

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

(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 $t \geq 1$

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

(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.

REGRET ANALYSIS OF FTRL: A HELPFUL LEMMA

- **Lemma:** Let $a_1^{\mathrm{FTRL}}, a_2^{\mathrm{FTRL}}, \ldots$, be the sequence of actions 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

$$R_T^{\mathrm{FTRL}}(\tilde{a}) = \sum_{t=1}^{T} \left(\ell(a_t^{\mathrm{FTRL}}, z_t) - \ell(\tilde{a}, z_t) \right)$$

$$\leq \psi(\tilde{a}) - \psi(a_1^{\mathrm{FTRL}}) + \sum_{t=1}^{T} \left(\ell(a_t^{\mathrm{FTRL}}, z_t) - \ell(a_{t+1}^{\mathrm{FTRL}}, z_t) \right).$$

REGRET ANALYSIS OF FTRL: A HELPFUL LEMMA

- **Lemma:** Let $a_1^{\mathrm{FTRL}}, a_2^{\mathrm{FTRL}}, \ldots$ be the sequence of actions 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

$$R_T^{\mathrm{FTRL}}(\tilde{a}) = \sum_{t=1}^{T} \left(\ell(a_t^{\mathrm{FTRL}}, z_t) - \ell(\tilde{a}, z_t) \right)$$

$$\leq \psi(\tilde{a}) - \psi(a_1^{\mathrm{FTRL}}) + \sum_{t=1}^{T} \left(\ell(a_t^{\mathrm{FTRL}}, z_t) - \ell(a_{t+1}^{\mathrm{FTRL}}, z_t) \right).$$

- *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 $(a, z) = a^\top z$ for online linear optimization (OLO) problems.
- For this purpose, the squared L2-norm regularization will be used:

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

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 $(a, z) = a^\top z$ for online linear optimization (OLO) problems.
- For this purpose, the squared L2-norm regularization will be used:

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

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

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

FTRL FOR ONLINE LINEAR OPTIMIZATION

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

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

  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

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

FTRL FOR ONLINE LINEAR OPTIMIZATION

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

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

  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

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

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

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

- 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:

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

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:**

Reminder (1): $R_T^{\mathrm{FTRL}}(\bar{a}) \leq \psi(\bar{a}) - \psi(a_1^{\mathrm{FTRL}}) + \sum_{t=1}^{T} \Big(L(a_t^{\mathrm{FTRL}}, z_t) - L(a_{t+1}^{\mathrm{FTRL}}, z_t) \Big)$.

Reminder (2): $a_{t+1}^{\mathrm{FTRL}} = a_t^{\mathrm{FTRL}} - \eta\, z_t, \quad t = 1, \dots, T-1.$

- For sake of brevity, we write $a_1, a_2, \dots$ for $a_1^{\mathrm{FTRL}}, a_2^{\mathrm{FTRL}}, \dots$

FTRL FOR OLO: THEORETICAL GUARANTEES

- **Proof:**

Reminder (1): $R_T^{\mathrm{FTRL}}(\tilde{a}) \leq \psi(\tilde{a}) - \psi(a_1^{\mathrm{FTRL}}) + \sum_{t=1}^{T} \left(L(a_t^{\mathrm{FTRL}}, z_t) - L(a_{t+1}^{\mathrm{FTRL}}, z_t) \right).$
Reminder (2): $a_{t+1}^{\mathrm{FTRL}} = a_t^{\mathrm{FTRL}} - \eta\, z_t, \quad t = 1, \ldots, T-1.$

- For sake of brevity, we write $a_1, a_2, \ldots$ for $a_1^{\mathrm{FTRL}}, a_2^{\mathrm{FTRL}}, \ldots$
- With this,

$$
\begin{aligned}
R_T^{FTRL}(\tilde{a}) &\leq \psi(\tilde{a}) - \psi(a_1) + \sum_{t=1}^{T} \left(L(a_t, z_t) - L(a_{t+1}, z_t) \right) && \text{(Reminder (1))}\\
&\leq \frac{1}{2\eta} \|\tilde{a}\|_2^2 + \sum_{t=1}^{T} \left(a_t^\top z_t - a_{t+1}^\top z_t \right) && (\psi(a_1) \geq 0 \text{ and definition of } \psi)\\
&= \frac{1}{2\eta} \|\tilde{a}\|_2^2 + \sum_{t=1}^{T} \left(a_t^\top - a_{t+1}^\top \right) z_t && \text{(Distributivity)}\\
&= \frac{1}{2\eta} \|\tilde{a}\|_2^2 + \eta \sum_{t=1}^{T} \|z_t\|_2^2 . && \text{(Reminder (2))}
\end{aligned}
$$

$\square$

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 the advantages of FTRL for OLO and OQO problems:

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