FOLLOW THE REGULARIZED LEADER

- To overcome the shortcomings of the FTL algorithm, one can incorporate a regularization function ψ : A → R₊ into the action choice of FTL, which leads to more stability.
- To be more precise, let for t ≥ 1

$$a\overset{\text{FTRL}}{a} \in \underset{a \in \mathcal{A}}{\operatorname{arg min}} \left(\psi(a) + \sum_{s=1}^{t-1} (a, z_s) \right),$$

then the algorithm choosing a FTRL in time step Fis 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 ψ : A → R₊ into the action choice of FTL, which leads to more stability.
- To be more precise, let for t ≥ 1

$$a\overset{\text{FTRL}}{a} \in \underset{a \in \mathcal{A}}{\operatorname{arg min}} \left(\psi(a) + \sum_{s=1}^{t-1} (a, z_s) \right),$$

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

Interpretation: The algorithm predicts a_t as the element in 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 ψ : A → R₊ into the action choice of FTL, which leads to more stability.
- To be more precise, let for t ≥ 1

$$a\overset{\text{FTRL}}{a} \in \underset{a \in \mathcal{A}}{\operatorname{arg min}} \left(\psi(a) + \sum_{s=1}^{t-1} (a, z_s) \right),$$

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

- Interpretation: The algorithm predicts a_t as the element in 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 ψ ≡ 0, then FTRL equals FTL:



REGRET ANALYSIS OF FTRL: A HELPFUL LEMMA

Lemma: Let a₁^{FTRL}, a₂ a₂^{FTRL}, be be the sequence of actions coming used by the FTRL algorithm for the environmental data sequence z₁, z₂,
 Then, for all ã ∈ A we have

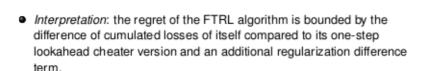
$$\begin{split} R_{T^{1T}}^{\mathrm{FTRL}}(\tilde{\boldsymbol{a}}) &= \sum_{t=1}^{r} \left((\boldsymbol{a}_{t^{2}t}^{\mathrm{FTRL}}, \boldsymbol{z}_{t}) - (\tilde{\boldsymbol{a}}_{t^{2}t^{2}}) \right) \\ &\leq \psi(\tilde{\boldsymbol{a}}) - \psi(\boldsymbol{a}_{t}^{\mathrm{FTRL}}) + \sum_{t=t+1}^{TT} \left((\boldsymbol{a}_{t}^{\mathrm{FTRL}}, \boldsymbol{z}_{t}^{2}\boldsymbol{z}_{t}) + ((\boldsymbol{a}_{t+1}^{\mathrm{FTRL}}, \boldsymbol{z}_{t}^{2}\boldsymbol{z}_{t}) \right). \end{split}$$



REGRET ANALYSIS OF FTRL: A HELPFUL LEMMA

 Lemma: Let a₁^{FTRL}, a₂^{ETRL}, he be the sequence of actions coming used by the FTRL algorithm for the environmental data sequence z₁, z₂,
 Then, for all ã ∈ A we have

$$\begin{split} R_{T^{1,t}}^{\text{F-IRIL}}(\tilde{\boldsymbol{a}}) &= \sum_{t=1}^{T} \left((\boldsymbol{a}_{t^{2}t}^{\text{F-TRIL}}, \boldsymbol{z}_{t}) - (\tilde{\boldsymbol{a}}_{t}^{\text{F-JRIL}}) \right) \\ &\leq \psi(\tilde{\boldsymbol{a}}) - \psi(\boldsymbol{a}_{t}^{\text{F-TRIL}}) + \sum_{t=t+1}^{TT} \left((\boldsymbol{a}_{t}^{\text{F-TRIL}}, \boldsymbol{z}_{t}^{\text{F-JR-L}}, \boldsymbol{z}_{t}^{\text{F-JR-L}}) + ((\boldsymbol{a}_{t+1}^{\text{F-TR-L}}, \boldsymbol{z}_{t}^{\text{F-JR-L}}, \boldsymbol{z}_{t}^{\text{F-JR-L}}) \right). \end{split}$$





(The proof is similar.)



- In the following, we analyze the FTRL algorithm for the linear loss
 (a₃z) = a₃z for online linear optimization (QLO) problemss.
- 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.



- In the following, we analyze the FTRL algorithm for the linear loss
 (a;z) = a z for online linear optimization (QLO) problemss.
- 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

$$a_{t_t}^{\text{ETRL}} = -\eta \sum_{s=1}^{t-1} z_s.$$



- In the following, we analyze the FTRL algorithm for the linear loss
 (a;z) = a z for online linear optimization (QLO) problemss.
- 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

$$a_{t_l}^{\text{ETRL}} = -\eta \sum_{s=1}^{t-1} Z_s.$$

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

$$a_{t+1}^{\text{FTRL}} = a_{t}^{\text{FftRL}} - \eta \eta z_{t}, \quad t = 1, ..., T - 1.$$



- In the following, we analyze the FTRL algorithm for the linear loss
 (asz) = a z for online linear optimization (QLO) problemss.
- 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

$$a_{t_t}^{\text{ETRL}} = -\eta \sum_{s=1}^{t-1} Z_s.$$

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

$$a_{t^2t^2t^{-1}}^{FFRIL} \equiv a_t^{FfIRL} - \gamma_1 \eta_1 z_t, \quad t \stackrel{\leftarrow}{\leftarrow} 1, \dots, TT-1.1.$$

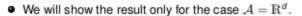
Interpretation: $-z_l$ is the direction in which the update of a_l^{FTRL} to a_{l+1}^{FTRL} is conducted with step size η in order to reduce the loss.



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

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



 For the more general case, where A is a strict subset of R^d, we need a slight modification of the update formula above:

$$a\overline{\mathbf{a}}_{t}^{\text{ECRL}} = \prod_{A} \left(\gamma - \sum_{i=1}^{t-1} Z_{i} \right) = \underset{\mathbf{a} \in A}{\operatorname{arg min}} \left\| \mathbf{a} - \eta \sum_{i=1}^{t+1} Z_{i} \right\|_{22}^{22}.$$

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):
$$R_T^{\text{FTRL}}(\tilde{\mathbf{a}}) \leq \psi(\tilde{\mathbf{a}}) + \psi(\tilde{\mathbf{a}}_1^{\text{ETRL}}) + \sum_{t=1}^T \left((\tilde{\mathbf{a}}_t^{\text{FTRL}}, z_t) (\tilde{\mathbf{a}}_1^{\text{FTRL}}, z_t) \right)$$
.

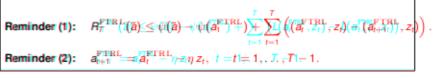
Reminder (2): $a_{t+1}^{\text{FTRL}} = \tilde{a}_t^{\text{ETRL}} + z_t \eta z_t$, $t = t = 1, ..., T = 1$.



For sake of brevity, we write a₁, a₂, ... for a₁ TRL TRL

FTRL FOR OLO: THEORETICAL GUARANTEES

Proof:





- For sake of brevity, we write a₁, a₂,... for a₁^{PTRL} a₂^{PTRL}....
- · With this.

$$\begin{split} R_T^{FTRL}(\ddot{a}) & \leq \psi(\ddot{a}) - \psi(a_1) + \sum_{t=1}^{T} ((\underbrace{a_{ij} z_t}) - (\underbrace{a_{i'j} z_t})_t)) & \text{(Reminder (1))} \\ & \leq \frac{1}{2\eta} \|\ddot{a}\|_2^2 + \sum_{t=1}^{\bar{T}} (a_t^\top z_t - a_{t+1}^\top z_t) & (\psi(a_1) \geq 0 \text{ and definition of } \psi) \\ & = \frac{1}{2\eta} \|\ddot{a}\|_2^2 + \sum_{t=1}^{T} (a_t^\top - a_{t+1}^\top) z_t & \text{(Distributivity)} \\ & = \frac{1}{2\eta} \|\ddot{a}\|_2^2 + \eta \sum_{t=1}^{T} ||z_t||_2^2. & \text{(Reminder (2))} \end{split}$$

DESIRED RESULTS

- With the FTRL algorithm we can cope with
 - online quadratic optimization (OQO) problems by using no regularity (ψ ≡ 0). In this case, we have satisfactory regret guarantees and also a quick update rule for a^{FTIRI}_{t+1} (Itsis 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 (ψ ≡ 0). In this case, we have satisfactory regret guarantees and also a quick update rule for a^{F/T,RL}[I(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 (ψ ≡ 0). In this case, we have satisfactory regret guarantees and also a quick update rule for a^{FTIRI}_{t+1} (Itsis 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.
- 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 OCO problems:
 - (a) reasonable regret upper bounds;
 - (b) a quick update formula.

