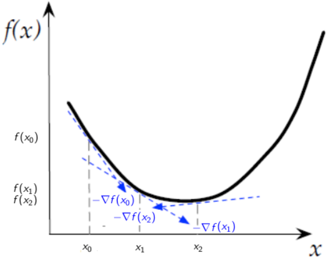
Advanced Machine Learning

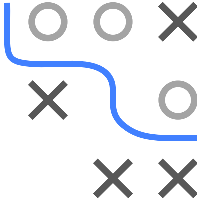
Online Convex Optimization - Part 1



Learning goals

- Get to know the class of online convex optimization problems
- Derive the online gradient descent as a suitable learning algorithm for such cases

ONLINE CONVEX OPTIMIZATION



- One of the most relevant instantiations of the online learning problem is the problem of *online convex optimization* (OCO), which is characterized by a loss function

$$: \mathcal{A} \times \mathcal{Z} \to \mathbb{R},$$

which is convex w.r.t. the action, i.e., $a \mapsto (a, z)$ is convex for any $z \in \mathcal{Z}$.

ONLINE CONVEX OPTIMIZATION

- One of the most relevant instantiations of the online learning problem is the problem of *online convex optimization* (OCO), which is characterized by a loss function

$$: \mathcal{A} \times \mathcal{Z} \to \mathbb{R},$$

  which is convex w.r.t. the action, i.e., $a \mapsto (a, z)$ is convex for any $z \in \mathcal{Z}$.

- Note that both OLO and OQO belong to the class of online convex optimization problems:
  - *Online linear optimization (OLO) with convex action spaces:*

$$(a, z) = a^\top z$$

    is a convex function in $a \in \mathcal{A}$, provided $\mathcal{A}$ is convex.
  - *Online quadratic optimization (OQO) with convex action spaces:*

$$(a, z) = \frac{1}{2} ||a - z||_2^2$$

    is a convex function in $a \in \mathcal{A}$, provided $\mathcal{A}$ is convex.

ONLINE GRADIENT DESCENT: MOTIVATION

- We have seen that the FTRL algorithm with the $_2$ norm regularization $\psi(a) = \frac{1}{2\eta} \left|\left|a\right|\right|_2^2$ achieves satisfactory results for online linear optimization (OLO) problems, that is, if $(a, z) = L^{\mathrm{lin}}(a, z) := a^\top z$, then we have
  - *Fast updates* — If $\mathcal{A} = \mathbb{R}^d$, then

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

  - *Regret bounds* — By an appropriate choice of η and some (mild) assumptions on $\mathcal{A}$ and $\mathcal{Z}$, we have

$$R_T^{\mathrm{FTRL}} = o(T).$$

ONLINE GRADIENT DESCENT: MOTIVATION

Apparently, the nice form of the loss function L^{lin} is responsible for the appealing properties of FTRL in this case. Indeed, since $\nabla_a L^{\mathrm{lin}}(a, z) = z$ note that the update rule can be written as

$$a_{t+1}^{\mathrm{FTRL}} = a_t^{\mathrm{FTRL}} - \eta\, z_t = a_t^{\mathrm{FTRL}} - \eta\, \nabla_a L^{\mathrm{lin}}(a_t^{\mathrm{FTRL}}, z_t).$$

ONLINE GRADIENT DESCENT: MOTIVATION

Apparently, the nice form of the loss function L^{lin} is responsible for the appealing properties of FTRL in this case. Indeed, since $\nabla_a L^{\mathrm{lin}}(a, z) = z$ note that the update rule can be written as

$$a_{t+1}^{\mathrm{FTRL}} = a_t^{\mathrm{FTRL}} - \eta\, z_t = a_t^{\mathrm{FTRL}} - \eta\, \nabla_a L^{\mathrm{lin}}(a_t^{\mathrm{FTRL}}, z_t).$$

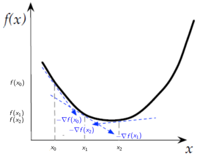
Interpretation: In each time step $t+1$, we are following the direction with the steepest decrease of the most recent loss (represented by $-\nabla L^{\mathrm{lin}}(a_t^{\mathrm{FTRL}}, z_t)$) from the current "position" a_t^{FTRL} with the step size η

ONLINE GRADIENT DESCENT: MOTIVATION

Apparently, the nice form of the loss function L^{lin} is responsible for the appealing properties of FTRL in this case. Indeed, since $\nabla_a L^{\mathrm{lin}}(a, z) = z$ note that the update rule can be written as

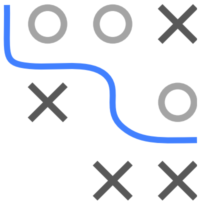
$$a_{t+1}^{\mathrm{FTRL}} = a_t^{\mathrm{FTRL}} - \eta\, z_t = a_t^{\mathrm{FTRL}} - \eta\, \nabla_a L^{\mathrm{lin}}(a_t^{\mathrm{FTRL}}, z_t).$$

Interpretation: In each time step $t+1$, we are following the direction with the steepest decrease of the most recent loss (represented by $-\nabla L^{\mathrm{lin}}(a_t^{\mathrm{FTRL}}, z_t)$) from the current "position" a_t^{FTRL} with the step size η



$\Rightarrow$ Gradient Descent.

ONLINE GRADIENT DESCENT: MOTIVATION



- **Question:** How to transfer this idea of the Gradient Descent for the update formula to other loss functions, while still preserving the regret bounds?

ONLINE GRADIENT DESCENT: MOTIVATION

- **Question:** How to transfer this idea of the Gradient Descent for the update formula to other loss functions, while still preserving the regret bounds?
- **Solution (for convex losses):** Recall the equivalent characterization of convexity of differentiable convex functions:

$$
\begin{aligned}
f : S \to \mathbb{R} \text{ is convex } &\Leftrightarrow f(y) \geq f(x) + (y - x)^\top \nabla f(x) \text{ for any } x, y \in S \\
&\Leftrightarrow f(x) - f(y) \leq (x - y)^\top \nabla f(x) \text{ for any } x, y \in S.
\end{aligned}
$$

ONLINE GRADIENT DESCENT: MOTIVATION

- **Question:** How to transfer this idea of the Gradient Descent for the update formula to other loss functions, while still preserving the regret bounds?
- **Solution (for convex losses):** Recall the equivalent characterization of convexity of differentiable convex functions:

$$
\begin{aligned}
f : S \to \mathbb{R} \text{ is convex } &\Leftrightarrow f(y) \geq f(x) + (y - x)^\top \nabla f(x) \text{ for any } x, y \in S \\
&\Leftrightarrow f(x) - f(y) \leq (x - y)^\top \nabla f(x) \text{ for any } x, y \in S.
\end{aligned}
$$

- This means if we are dealing with a loss function $: \mathcal{A} \times \mathcal{Z} \to \mathbb{R}$, which is convex and differentiable in its first argument ($\mathcal{A}$ has also to be convex), then

$$
(a, z) - (\tilde{a}, z) \leq (a - \tilde{a})^\top \nabla_a(a, z), \quad \forall a, \tilde{a} \in \mathcal{A}, z \in \mathcal{Z}.
$$

ONLINE GRADIENT DESCENT: MOTIVATION

Reminder: $(a, z) - (\tilde{a}, z) \leq (a - \tilde{a})^\top \nabla_a(a, z), \quad \forall a, \tilde{a} \in \mathcal{A}, z \in \mathcal{Z}.$

ONLINE GRADIENT DESCENT: MOTIVATION

Reminder: $(a, z) - (\tilde{a}, z) \leq (a - \tilde{a})^\top \nabla_a(a, z), \quad \forall a, \tilde{a} \in \mathcal{A}, z \in \mathcal{Z}.$

- Let $z_1, \ldots, z_T$ arbitrary environmental data and $a_1, \ldots, a_T$ be some arbitrary action sequence. Substitute $\tilde{z}_t := \nabla_a(a_t, z_t)$ and note that

ONLINE GRADIENT DESCENT: MOTIVATION

Reminder: $(a, z) - (\tilde{a}, z) \leq (a - \tilde{a})^\top \nabla_a(a, z), \quad \forall a, \tilde{a} \in \mathcal{A}, z \in \mathcal{Z}.$

- Let $z_1, \ldots, z_T$ arbitrary environmental data and $a_1, \ldots, a_T$ be some arbitrary action sequence. Substitute $\tilde{z}_t := \nabla_a(a_t, z_t)$ and note that

$$
\begin{aligned}
R_T(\tilde{a}) &= \sum_{t=1}^{T}(a_t, z_t) - (\tilde{a}, z_t) \leq \sum_{t=1}^{T}(a_t - \tilde{a})^\top \nabla_a(a_t, z_t) \\
&= \sum_{t=1}^{T}(a_t - \tilde{a})^\top \tilde{z}_t = \sum_{t=1}^{T} a_t^\top \tilde{z}_t - \tilde{a}^\top \tilde{z}_t = \sum_{t=1}^{T} L^{\mathrm{lin}}(a_t, \tilde{z}_t) - L^{\mathrm{lin}}(\tilde{a}, \tilde{z}_t).
\end{aligned}
$$

ONLINE GRADIENT DESCENT: MOTIVATION

Reminder: $(a, z) - (\tilde{a}, z) \leq (a - \tilde{a})^\top \nabla_a(a, z), \quad \forall a, \tilde{a} \in \mathcal{A}, z \in \mathcal{Z}.$

- Let $z_1, \ldots, z_T$ arbitrary environmental data and $a_1, \ldots, a_T$ be some arbitrary action sequence. Substitute $\tilde{z}_t := \nabla_a(a_t, z_t)$ and note that

$$
\begin{aligned}
R_T(\tilde{a}) &= \sum_{t=1}^{T} (a_t, z_t) - (\tilde{a}, z_t) \leq \sum_{t=1}^{T} (a_t - \tilde{a})^\top \nabla_a(a_t, z_t) \\
&= \sum_{t=1}^{T} (a_t - \tilde{a})^\top \tilde{z}_t = \sum_{t=1}^{T} a_t^\top \tilde{z}_t - \tilde{a}^\top \tilde{z}_t = \sum_{t=1}^{T} L^{\mathrm{lin}}(a_t, \tilde{z}_t) - L^{\mathrm{lin}}(\tilde{a}, \tilde{z}_t).
\end{aligned}
$$

Conclusion: The regret of a learner with respect to a differentiable and convex loss function is bounded by the regret corresponding to an online linear optimization problem with environmental data $\tilde{z}_t = \nabla_a(a_t, z_t)$.

ONLINE GRADIENT DESCENT: MOTIVATION

Reminder: $(a, z) - (\tilde{a}, z) \leq (a - \tilde{a})^\top \nabla_a(a, z), \quad \forall a, \tilde{a} \in \mathcal{A}, z \in \mathcal{Z}.$

- Let $z_1, \ldots, z_T$ arbitrary environmental data and $a_1, \ldots, a_T$ be some arbitrary action sequence. Substitute $\tilde{z}_t := \nabla_a(a_t, z_t)$ and note that

$$
\begin{aligned}
R_T(\tilde{a}) &= \sum_{t=1}^{T} (a_t, z_t) - (\tilde{a}, z_t) \leq \sum_{t=1}^{T} (a_t - \tilde{a})^\top \nabla_a(a_t, z_t) \\
&= \sum_{t=1}^{T} (a_t - \tilde{a})^\top \tilde{z}_t = \sum_{t=1}^{T} a_t^\top \tilde{z}_t - \tilde{a}^\top \tilde{z}_t = \sum_{t=1}^{T} L^{\mathrm{lin}}(a_t, \tilde{z}_t) - L^{\mathrm{lin}}(\tilde{a}, \tilde{z}_t).
\end{aligned}
$$

Conclusion: The regret of a learner with respect to a differentiable and convex loss function is bounded by the regret corresponding to an online linear optimization problem with environmental data $\tilde{z}_t = \nabla_a(a_t, z_t)$.

- **We know:** Online linear optimization problems can be tackled by means of the FTRL algorithm!

ONLINE GRADIENT DESCENT: MOTIVATION

Reminder: $(a, z) - (\tilde{a}, z) \leq (a - \tilde{a})^\top \nabla_a(a, z), \quad \forall a, \tilde{a} \in \mathcal{A}, z \in \mathcal{Z}.$

- Let $z_1, \ldots, z_T$ arbitrary environmental data and $a_1, \ldots, a_T$ be some arbitrary action sequence. Substitute $\tilde{z}_t := \nabla_a(a_t, z_t)$ and note that

$$
\begin{aligned}
R_T(\tilde{a}) &= \sum_{t=1}^{T} (a_t, z_t) - (\tilde{a}, z_t) \leq \sum_{t=1}^{T} (a_t - \tilde{a})^\top \nabla_a(a_t, z_t) \\
&= \sum_{t=1}^{T} (a_t - \tilde{a})^\top \tilde{z}_t = \sum_{t=1}^{T} a_t^\top \tilde{z}_t - \tilde{a}^\top \tilde{z}_t = \sum_{t=1}^{T} L^{\mathrm{lin}}(a_t, \tilde{z}_t) - L^{\mathrm{lin}}(\tilde{a}, \tilde{z}_t).
\end{aligned}
$$

Conclusion: The regret of a learner with respect to a differentiable and convex loss function is bounded by the regret corresponding to an online linear optimization problem with environmental data $\tilde{z}_t = \nabla_a(a_t, z_t)$.

- **We know:** Online linear optimization problems can be tackled by means of the FTRL algorithm!

$\rightsquigarrow$ Incorporate the substitution $\tilde{z}_t = \nabla_a(a_t, z_t)$ into the update formula of FTRL with squared L2-norm regularization.

ONLINE GRADIENT DESCENT: DEFINITION

- The corresponding algorithm which chooses its action according to these considerations is called the *Online Gradient Descent* (OGD) algorithm with step size $\eta > 0$. It holds in particular,

$$a_{t+1}^{\mathrm{OGD}} = a_t^{\mathrm{OGD}} - \eta \nabla_a (a_t^{\mathrm{OGD}}, z_t), \quad t = 1, \ldots T. \tag{1}$$

(Technical side note: For this update formula we assume that $\mathcal{A} = \mathbb{R}^d$. Moreover, the first action a_1^{OGD} is arbitrary.)