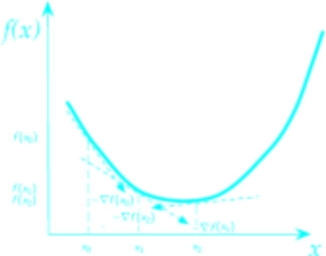
ONLINE CONVEX OPTIMIZATION

- 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 $z \in \mathcal{Z}$.

Advanced Machine Learning

Online Convex Optimization



Learning goals

- Get to know the class of online convex optimization problems
- See the online gradient descent as a satisfactory learning algorithm for such cases
- Know its connection to the FTRL via linearization of convex functions

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 $L: \mathcal{A} \times \mathcal{Z} \to \mathbb{R}$, 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 $\psi(a) = \frac{1}{2\eta}\|a\|_2^2$ achieves satisfactory results for online linear optimization (OLO) problems, that is, if $(a, z) = L^{\mathtt{lin}}(a, z) := a^\top z$, 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 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 $L : \mathcal{A} \times \mathcal{Z} \to \mathbb{R}$, which is convex w.r.t. the action, i.e., $a \mapsto L(a, z)$ 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:* $L(a, z) = a^\top z$ is a convex function in $a \in \mathcal{A}$, provided $\mathcal{A}$ is convex.
  - *Online quadratic optimization (OQO) with convex action spaces:* $L(a, z) = \frac{1}{2}\|a - z\|_2^2$ is a convex function in $a \in \mathcal{A}$, provided $\mathcal{A}$ is convex.

ONLINE GRADIENT DESCENT: MOTIVATION

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

- We have seen that the FTRL algorithm with the L_2 norm regularization $\psi(a) = \frac{1}{2\eta}\|a\|^2$ achieves satisfactory results for online linear optimization (OLO) problems, that is, if $L(a, z) = L^{\mathtt{lin}}(a, z) := a^\top z$, then we have $a_{t+1}^{\mathtt{FTRL}} = a_t^{\mathtt{FTRL}} - \eta\, z_t = a_t^{\mathtt{FTRL}} - \eta\, \nabla_a L^{\mathtt{lin}}(a_t^{\mathtt{FTRL}}, z_t)$.
  - *Fast updates* — If $\mathcal{A} = \mathbb{R}^d$, then
  $$a_{t+1}^{\mathtt{FTRL}} = a_t^{\mathtt{FTRL}} - \eta\, z_t, \qquad t = 1, \ldots, T;$$
  - *Regret bounds* — By an appropriate choice of η and some (mild) assumptions on $\mathcal{A}$ and $\mathcal{Z}$, we have
  $$R_T^{\mathtt{FTRL}} = o(T).$$

ONLINE GRADIENT DESCENT: MOTIVATION

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

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

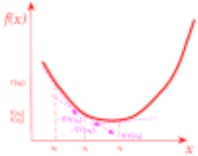
Interpretation: In each time step $t+1$, we are following the direction with the steepest decrease of the loss (represented by $-\nabla L^{\mathtt{lin}}(a_t^{\mathrm{FTRL}}, z_t)$) from the current "position" a_t^{FTRL} with the step size η

ONLINE GRADIENT DESCENT: MOTIVATION

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

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

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



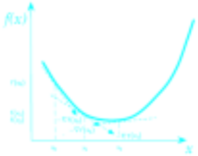
⇒ Gradient Descent.

ONLINE GRADIENT DESCENT: MOTIVATION

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

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

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



⇒ Gradient Descent.

- **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:** $(a, z) - (\tilde{a}, z) \leq (a - \tilde{a})^\top \nabla_a(a, z), \quad \forall a, \tilde{a} \in \mathcal{A}, z \in \mathcal{Z}.$
- **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 $L : \mathcal{A} \times \mathcal{Z} \to \mathbb{R}$, which is convex and differentiable in its first argument ($\mathcal{A}$ has also to be convex), then

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

ONLINE GRADIENT DESCENT: MOTIVATION

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

$$R_T(\tilde{a}) = \sum_{t=1}^{T} (a_t, z_t) - (\tilde{a}, z_t) \leq \sum_{t=1}^{T} (a_t - \tilde{a})^\top \nabla_a (a_t, z_t)$$

$$= \sum_{t=1}^{T} (a_t - \tilde{a})^\top \tilde{z}_t = \sum_{t=1}^{T} a_t^\top \tilde{z}_t - \tilde{a}^\top \tilde{z}_t = \sum_{t=1}^{T} L^{\text{lin}}(a_t, \tilde{z}_t) - L^{\text{lin}}(\tilde{a}, \tilde{z}_t).$$

ONLINE GRADIENT DESCENT: MOTIVATION

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

$$R_T(\tilde{a}) = \sum_{t=1}^{T} (a_t, z_t) - (\tilde{a}, z_t) \leq \sum_{t=1}^{T} (a_t - \tilde{a})^\top \nabla_a (a_t, z_t)$$

$$= \sum_{t=1}^{T} (a_t - \tilde{a})^\top \tilde{z}_t = \sum_{t=1}^{T} a_t^\top \tilde{z}_t - \tilde{a}^\top \tilde{z}_t = \sum_{t=1}^{T} L^{\text{lin}}(a_t, \tilde{z}_t) - L^{\text{lin}}(\tilde{a}, \tilde{z}_t).$$

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

$$R_T(\tilde{a}) = \sum_{t=1}^{T} (a_t, z_t) - (\tilde{a}, z_t) \leq \sum_{t=1}^{T} (a_t - \tilde{a})^\top \nabla_a (a_t, z_t)$$

$$= \sum_{t=1}^{T} (a_t - \tilde{a})^\top \tilde{z}_t = \sum_{t=1}^{T} a_t^\top \tilde{z}_t - \tilde{a}^\top \tilde{z}_t = \sum_{t=1}^{T} L^{\text{lin}}(a_t, \tilde{z}_t) - L^{\text{lin}}(\tilde{a}, \tilde{z}_t).$$

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

$$R_T(\tilde{a}) = \sum_{t=1}^{T} (a_t, z_t) - (\tilde{a}, z_t) \leq \sum_{t=1}^{T} (a_t - \tilde{a})^{\top} \nabla_a (a_t, z_t)$$

$$= \sum_{t=1}^{T} (a_t - \tilde{a})^{\top} \tilde{z}_t = \sum_{t=1}^{T} a_t^{\top} \tilde{z}_t - \tilde{a}^{\top} \tilde{z}_t = \sum_{t=1}^{T} L^{\mathtt{lin}}(a_t, \tilde{z}_t) - L^{\mathtt{lin}}(\tilde{a}, \tilde{z}_t).$$

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!

⇝ 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 AND PROPERTIES

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

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

$$R_T(\tilde{a}) = \sum_{t=1}^{T} L(a_t, z_t) - L(\tilde{a}, z_t) \le \sum_{t=1}^{T} (a_t - \tilde{a})^\top \nabla_a L(a_t, z_t)$$

$$\sum_{t=1}^{T} (a_t - \tilde{a})^\top \tilde{z}_t = \sum_{t=1}^{T} a_t^\top \tilde{z}_t - \tilde{a}^\top \tilde{z}_t .$$

- 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,

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

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

Conclusion: The regret of a learner with respect to a differentiable and convex loss function L is bounded by

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

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

ONLINE GRADIENT DESCENT: DEFINITION AND PROPERTIES

- 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,

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

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

- We have the following connection between FTRL and OGD:
  - Let $\tilde{z}_t^{\mathrm{OGD}} := \nabla_a(a_t^{\mathrm{OGD}}, z_t)$ for any $t = 1, \ldots, T$.
  - The update formula for FTRL with $_2$ norm regularization for the linear loss $L^{\mathtt{lin}}$ and the environmental data $\tilde{z}_t^{\mathrm{OGD}}$ is

$$a_{t+1}^{\mathrm{FTRL}} = a_t^{\mathrm{FTRL}} - \eta \tilde{z}_t^{\mathrm{OGD}} = a_t^{\mathrm{FTRL}} - \eta \nabla_a(a_t^{\mathrm{OGD}}, z_t).$$

  - If we have that $a_1^{\mathrm{FTRL}} = a_1^{\mathrm{OGD}}$, then it iteratively follows that $a_{t+1}^{\mathrm{FTRL}} = a_{t+1}^{\mathrm{OGD}}$ for any $t = 1, \ldots, T$ in this case.

ONLINE GRADIENT DESCENT: DEFINITION AND PROPERTIES

- 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,

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

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

- We have the following connection between FTRL and OGD:
  - Let $\tilde{z}_t^{\mathrm{OGD}} := \nabla_a L(a_t^{\mathrm{OGD}}, z_t)$ for any $t = 1, \ldots, T$.
  - The update formula for FTRL with L_2 norm regularization for the linear loss L^{lin} and the environmental data $\tilde{z}_t^{\mathrm{OGD}}$ is

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

  - If we have that $a_1^{\mathrm{FTRL}} = a_1^{\mathrm{OGD}}$, then it iteratively follows that $a_{t+1}^{\mathrm{FTRL}} = a_{t+1}^{\mathrm{OGD}}$ for any $t = 1, \ldots, T$ in this case.

- With the deliberations above we can infer that

$$\begin{aligned} R_{T,L}^{\mathrm{OGD}}(\tilde{a} \mid (z_t)_t) &= \sum_{t=1}^{T} L(a_t^{\mathrm{OGD}}, z_t) - L(\tilde{a}, z_t) \\ &\leq \sum_{t=1}^{T} L^{\mathrm{lin}}(a_t^{\mathrm{OGD}}, \tilde{z}_t^{\mathrm{OGD}}) - L^{\mathrm{lin}}(\tilde{a}, \tilde{z}_t^{\mathrm{OGD}}) \\ &\overset{(\text{if } a_1^{\mathrm{OGD}} = a_1^{\mathrm{FTRL}})}{=} \sum_{t=1}^{T} L^{\mathrm{lin}}(a_t^{\mathrm{FTRL}}, \tilde{z}_t^{\mathrm{OGD}}) - L^{\mathrm{lin}}(\tilde{a}, \tilde{z}_t^{\mathrm{OGD}}) \\ &= R_{T,L^{\mathrm{lin}}}^{\mathrm{FTRL}}(\tilde{a} \mid (\tilde{z}_t^{\mathrm{OGD}})_t), \end{aligned}$$

where we write in the subscripts of the regret the corresponding loss function and also include the corresponding environmental data as a second argument in order to emphasize the connections.

ONLINE GRADIENT DESCENT: DEFINITION AND PROPERTIES

- With the deliberations above we can infer that

$$
\begin{aligned}
R^{\mathrm{OGD}}_{T,L}(\tilde{a} \mid (z_t)_t) &= \sum_{t=1}^{T} (a_t^{\mathrm{OGD}}, z_t) - (\tilde{a}, z_t) \\
&\leq \sum_{t=1}^{T} L^{\mathtt{lin}}(a_t^{\mathrm{OGD}}, \tilde{z}_t^{\mathrm{OGD}}) - L^{\mathtt{lin}}(\tilde{a}, \tilde{z}_t^{\mathrm{OGD}}) \\
&\overset{(\text{if } a_1^{\mathrm{OGD}} = a_1^{\mathrm{FTRL}})}{=} \sum_{t=1}^{T} L^{\mathtt{lin}}(a_t^{\mathrm{FTRL}}, \tilde{z}_t^{\mathrm{OGD}}) - L^{\mathtt{lin}}(\tilde{a}, \tilde{z}_t^{\mathrm{OGD}}) \\
&= R^{\mathrm{FTRL}}_{T,L^{\mathtt{lin}}}(\tilde{a} \mid (\tilde{z}_t^{\mathrm{OGD}})_t),
\end{aligned}
$$

  where we write in the subscripts of the regret the corresponding loss function and also include the corresponding environmental data as a second argument in order to emphasize the connections.

- *Interpretation:* The regret of the FTRL algorithm (with $_2$ norm regularization) for the online linear optimization problem (characterized by the linear loss $L^{\mathtt{lin}}$) with environmental data $\tilde{z}_t^{\mathrm{OGD}}$ is an upper bound for the OGD algorithm for the online convex problem (characterized by a differentiable convex loss) with the original environmental data z_t.

ONLINE GRADIENT DESCENT: REGRET

- Due to this connection we immediately obtain a similar decomposition of the regret upper bound into a bias term and a variance term as for the FTRL algorithm for OLO problems.
- **Corollary.** Using the OGD algorithm on any online convex optimization problem (with differentiable loss function) leads to a regret of OGD with respect to any action $\tilde{a} \in \mathcal{A}$ of

$$R_T^{\mathrm{OGD}}(\tilde{a}) \le \frac{1}{2\eta}\|\tilde{a}\|_2^2 + \eta \sum_{t=1}^{T} \left\|\nabla_a(a_t^{\mathrm{OGD}}, z_t)\right\|_2^2 .$$

ONLINE GRADIENT DESCENT: TION AND PROPERTIES

- With the deliberations above we can infer that

$$R_{T,L}^{\mathrm{OGD}}(\tilde{a} \mid (z_t)) = \sum_{t=1}^{T} L(a_t^{\mathrm{OGD}}, z_t) - L(\tilde{a}, z_t)$$

$$\le \sum_{t=1}^{T} L^{\mathtt{lin}}(a_t^{\mathrm{OGD}}, \tilde{z}_t^{\mathrm{OGD}}) - L^{\mathtt{lin}}(\tilde{a}, \tilde{z}_t^{\mathrm{OGD}})$$

$$= \sum_{t=1}^{T} L^{\mathtt{lin}}(a_t^{\mathrm{FTRL}}, \tilde{z}_t^{\mathrm{OGD}}) - L^{\mathtt{lin}}(\tilde{a}, \tilde{z}_t^{\mathrm{OGD}})$$

$$= R_{T,L^{\mathtt{lin}}}^{\mathrm{FTRL}}(\tilde{a} \mid (\tilde{z}_t^{\mathrm{OGD}}))$$

where we write in the subscripts of the regret the corresponding loss function and also include the corresponding environmental data as a second argument in order to emphasize the connections.

- *Interpretation:* The regret of the FTRL algorithm (with L_2 norm regularization) for the online linear optimization problem (characterized by the linear loss $L^{\mathtt{lin}}$) with environmental data $\tilde{z}_t^{\mathrm{OGD}}$ is an upper bound for the OGD algorithm for the online convex problem (characterized by a differentiable convex loss L) with the original environmental data z_t.

ONLINE GRADIENT DESCENT: REGRET

- Due to this connection we immediately obtain a similar decomposition of the regret upper bound into a bias term and a variance term as for the FTRL algorithm for OLO problems.
- **Corollary.** Using the OGD algorithm on any online convex optimization problem (with differentiable loss function) leads to a regret of OGD with respect to any action $\tilde{a} \in \mathcal{A}$ of

$$
\begin{aligned}
R_T^{\text{OGD}}(\tilde{a}) &\leq \frac{1}{2\eta}\|\tilde{a}\|_2^2 + \eta \sum_{t=1}^{T} \|\tilde{z}_t^{\text{OGD}}\|_2^2 \\
&= \frac{1}{2\eta}\|\tilde{a}\|_2^2 + \eta \sum_{t=1}^{T} \|\nabla_a l(a_t^{\text{OGD}}, z_t)\|_2^2 .
\end{aligned}
$$

- Note that the step size $\eta > 0$ of OGD has the same role as the regularization magnitude of FTRL: It should balance the trade-off between the bias- and the variance-term.

ONLINE GRADIENT DESCENT: REGRET

- Due to this connection we immediately obtain a similar decomposition of the regret upper bound into a bias term and a variance term as for the FTRL algorithm for OLO problems.
- **Corollary:** Using the OGD algorithm on any online convex optimization problem (with differentiable loss function L) leads to a regret of OGD with respect to any action $\tilde{a} \in \mathcal{A}$ of

$$
\begin{aligned}
R_T^{\text{OGD}}(\tilde{a}) &\leq \frac{1}{2\eta}||\tilde{a}||_2^2 + \eta \sum\nolimits_{t=1}^{T} ||\tilde{z}_t^{\text{OGD}}||_2^2 \\
&= \frac{1}{2\eta}||\tilde{a}||_2^2 + \eta \sum\nolimits_{t=1}^{T} ||\nabla_a L(a_t^{\text{OGD}}, z_t)||_2^2 .
\end{aligned}
$$

- Note that the step size $\eta > 0$ of OGD has the same role as the regularization magnitude of FTRL: It should balance the trade-off between the bias- and the variance-term.

- As a consequence, we can also derive a similar order of the regret for the OGD algorithm on OCO problems as for the FTRL on OLO problems by imposing a slightly different assumption on the (new) “variance” term $\sum_{t=1}^{T} ||\nabla_a(a_t^{\text{OGD}}, z_t)||_2^2$.

ONLINE GRADIENT DESCENT: REGRET

- As a consequence, we can also derive a similar order of the regret for the OGD algorithm on OCO problems as for the FTRL on OLO problems by imposing a slightly different assumption on the (new) “variance” term $\sum_{t=1}^T \|\nabla_a(a_t^{\texttt{OGD}}, z_t)\|_2^2$.
- **Corollary:** Suppose we use the OGD algorithm on an online convex optimization problem with a convex action space $\mathcal{A} \subset \mathbb{R}^d$ such that
  - $\sup_{\tilde{a}\in\mathcal{A}} \|\tilde{a}\|_2 \leq B$ for some finite constant $B > 0$
  - $\sup_{a\in\mathcal{A}, z\in\mathcal{Z}} \|\nabla_a(a, z)\|_2 \leq V$ for some finite constant $V > 0$.

  Then, by choosing the step size η for OGD as $\eta = \frac{B}{V\sqrt{2T}}$ we get

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

REGRET LOWER BOUNDS FOR OCO

- **Theorem.** For any online learning algorithm there exists an online convex optimization problem characterized by a convex loss function, a bounded (convex) action space $\mathcal{A} = [-B, B]^d$ and bounded gradients $\sup_{a \in \mathcal{A}, z \in \mathcal{Z}} ||\nabla_a(a, z)||_2 \leq V$ for some finite constants $B, V > 0$, such that the algorithm incurs a regret of $\Omega(\sqrt{T})$ in the worst case.
- **Corollary:** Suppose we use the OGD algorithm on an online convex optimization problem with a convex action space $\mathcal{A} \subset \mathbb{R}^d$ such that
    - $\sup_{\tilde{a} \in \mathcal{A}} ||\tilde{a}||_2 \leq B$ for some finite constant $B > 0$
    - $\sup_{a \in \mathcal{A}, z \in \mathcal{Z}} ||\nabla_a L(a, z)||_2 \leq V$ for some finite constant $V > 0$.

  Then, by choosing the step size η for OGD as $\eta = \frac{B}{V\sqrt{2T}}$ we get

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

REGRET LOWER BOUNDS FOR OCO

- **Theorem.** For any online learning algorithm there exists an online convex optimization problem characterized by a convex loss function L, a bounded (convex) action space $\mathcal{A} = [-B, B]^d$ and bounded gradients $\sup_{a \in \mathcal{A}, z \in \mathcal{Z}} ||\nabla_a L(a, z)||_2 \leq V$ for some finite constants $B, V > 0$, such that the algorithm incurs a regret of $\Omega(\sqrt{T})$ in the worst case.
- Recall that under (almost) the same assumptions as the theorem above, we have $R_T^{\text{OGD}} \leq BV\sqrt{2T}$.

REGRET LOWER BOUNDS FOR OCO

- **Theorem.** For any online learning algorithm there exists an online convex optimization problem characterized by a convex loss function L, a bounded (convex) action space $\mathcal{A} = [-B, B]^d$ and bounded gradients $\sup_{a \in \mathcal{A}, z \in \mathcal{Z}} ||\nabla_a L(a, z)||_2 \leq V$ for some finite constants $B, V > 0$, such that the algorithm incurs a regret of $\Omega(\sqrt{T})$ in the worst case.
- Recall that under (almost) the same assumptions as the theorem above, we have $R_T^{\mathrm{OGD}} \leq BV\sqrt{2T}$.

$\leadsto$ This result shows that the Online Gradient Descent is *optimal* regarding its order of its regret with respect to the time horizon T.

REGRET LOWER BOUNDS FOR OCO

- **Theorem.** For any online learning algorithm there exists an online convex optimization problem characterized by a convex loss function L, a bounded (convex) action space $\mathcal{A} = [-B, B]^d$ and bounded gradients $\sup_{a\in\mathcal{A},z\in\mathcal{Z}} ||\nabla_a L(a, z)||_2 \leq V$ for some finite constants $B, V > 0$, such that the algorithm incurs a regret of $\Omega(\sqrt{T})$ in the worst case.

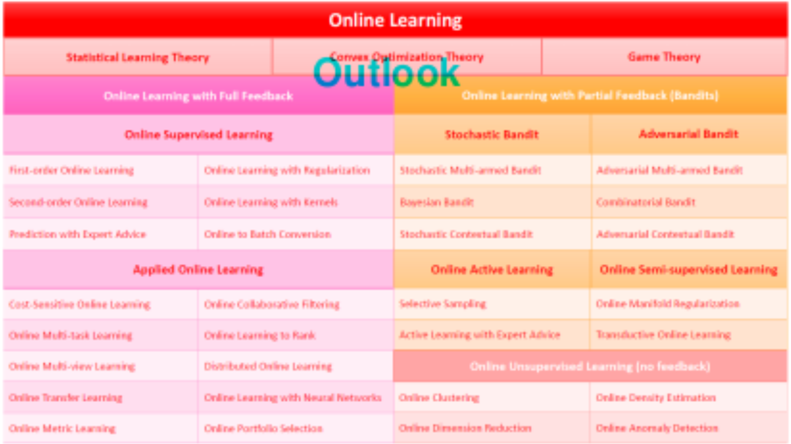
Outlook

- Recall that under (almost) the same assumptions as the theorem above, we have $R_T^{\mathrm{OGD}} \leq BV\sqrt{2T}$.

⇝ This result shows that the Online Gradient Descent is *optimal* regarding its order of its regret with respect to the time horizon T.

ONLINE MACHINE LEARNING: OUTLOOK

Online machine learning is a very large field of research.



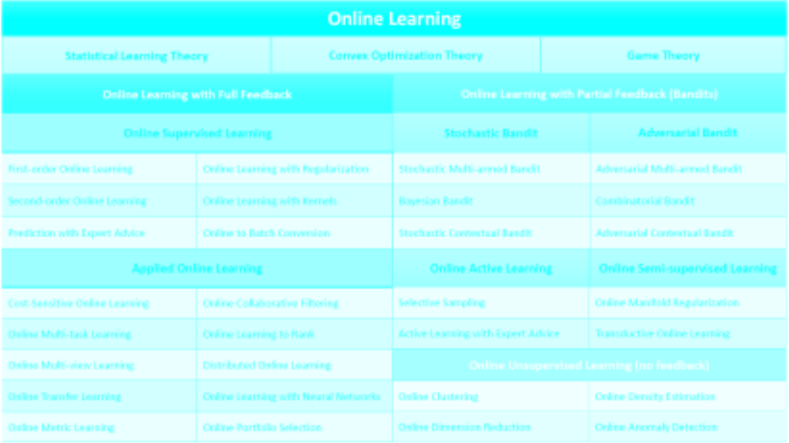
Outlook

<table>
<tr><th colspan="4">Online Learning</th></tr>
<tr><td>Statistical Learning Theory</td><td colspan="2">Convex Optimization Theory</td><td>Game Theory</td></tr>
<tr><td colspan="2">Online Learning with Full Feedback</td><td colspan="2">Online Learning with Partial Feedback (Bandits)</td></tr>
<tr><td colspan="2">Online Supervised Learning</td><td>Stochastic Bandit</td><td>Adversarial Bandit</td></tr>
<tr><td>First-order Online Learning</td><td>Online Learning with Regularization</td><td>Stochastic Multi-armed Bandit</td><td>Adversarial Multi-armed Bandit</td></tr>
<tr><td>Second-order Online Learning</td><td>Online Learning with Kernels</td><td>Bayesian Bandit</td><td>Combinatorial Bandit</td></tr>
<tr><td>Prediction with Expert Advice</td><td>Online to Batch Conversion</td><td>Stochastic Contextual Bandit</td><td>Adversarial Contextual Bandit</td></tr>
<tr><td colspan="2">Applied Online Learning</td><td>Online Active Learning</td><td>Online Semi-supervised Learning</td></tr>
<tr><td>Cost-Sensitive Online Learning</td><td>Online Collaborative Filtering</td><td>Selective Sampling</td><td>Online Manifold Regularization</td></tr>
<tr><td>Online Multi-task Learning</td><td>Online Learning to Rank</td><td>Active Learning with Expert Advice</td><td>Transductive Online Learning</td></tr>
<tr><td>Online Multi-view Learning</td><td>Distributed Online Learning</td><td colspan="2">Online Unsupervised Learning (no feedback)</td></tr>
<tr><td>Online Transfer Learning</td><td>Online Learning with Neural Networks</td><td>Online Clustering</td><td>Online Density Estimation</td></tr>
<tr><td>Online Metric Learning</td><td>Online Portfolio Selection</td><td>Online Dimension Reduction</td><td>Online Anomaly Detection</td></tr>
</table>

Figure: Hoi et al. (2018), "Online Learning: A Comprehensive Survey".

ONLINE MACHINE LEARNING: OUTLOOK

Online machine learning is a very large field of research.



Online Learning			
Statistical Learning Theory	Convex Optimization Theory	Game Theory	
Online Learning with Full Feedback		Online Learning with Partial Feedback (Bandits)	
Online Supervised Learning		Stochastic Bandit	Adversarial Bandit
First-order Online Learning	Online Learning with Regularization	Stochastic Multi-armed Bandit	Adversarial Multi-armed Bandit
Second-order Online Learning	Online Learning with Kernels	Bayesian Bandit	Combinatorial Bandit
Prediction with Expert Advice	Online to Batch Conversion	Stochastic Contextual Bandit	Adversarial Contextual Bandit
Applied Online Learning		Online Active Learning	Online Semi-supervised Learning
Cost-Sensitive Online Learning	Online Collaborative Filtering	Selective Sampling	Online Manifold Regularization
Online Multi-task Learning	Online Learning to Rank	Active Learning with Expert Advice	Transductive Online Learning
Online Multi-view Learning	Distributed Online Learning	Online Unsupervised Learning (no feedback)	
Online Transfer Learning	Online Learning with Neural Networks	Online Clustering	Online Density Estimation
Online Metric Learning	Online Portfolio Selection	Online Dimension Reduction	Online Anomaly Detection

Figure: Hoi et al. (2018), "Online Learning: A Comprehensive Survey".