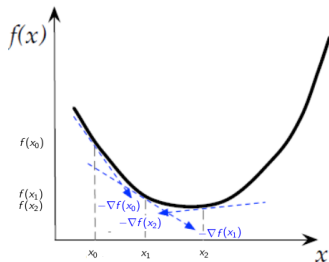


Advanced Machine Learning

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

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

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



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

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

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:*

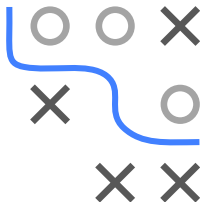
$$L(a, z) = a^\top z$$

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

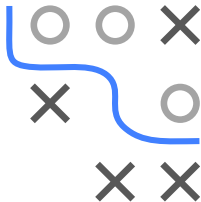
- We have seen that the FTRL algorithm with the L_2 norm regularization $\psi(a) = \frac{1}{2\eta} \|a\|_2^2$ achieves satisfactory results for online linear optimization (OLO) problems, that is, if $L(a, z) = L^{\text{lin}}(a, z) := a^\top z$, then we have

- *Fast updates* — If $\mathcal{A} = \mathbb{R}^d$, then

$$a_{t+1}^{\text{FTRL}} = a_t^{\text{FTRL}} - \eta z_t, \quad t = 1, \dots, T;$$

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

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



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 $\nabla_a L^{\text{lin}}(a, z) = z$ note that the update rule can be written as

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

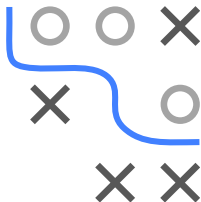


ONLINE GRADIENT DESCENT: MOTIVATION

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

$$\bar{a}_{t+1}^{\text{FTRL}} = \bar{a}_t^{\text{FTRL}} - \eta z_t = \bar{a}_t^{\text{FTRL}} - \eta \nabla_a L^{\text{lin}}(\bar{a}_t^{\text{FTRL}}, z_t).$$

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

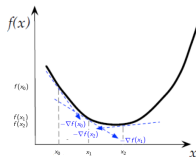


ONLINE GRADIENT DESCENT: MOTIVATION

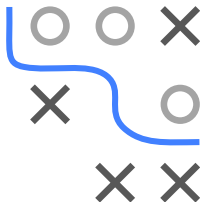
Apparently, the nice form of the loss function L^{lin} is responsible for the appealing properties of FTRL in this case. Indeed, since $\nabla_a L^{\text{lin}}(a, z) = z$ note that the update rule can be written as

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

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



⇒ Gradient Descent.



ONLINE GRADIENT DESCENT: MOTIVATION

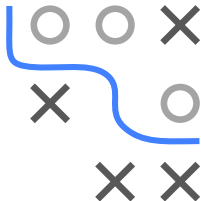
- **Question:** How to transfer this idea of the Gradient Descent for the update formula to other loss functions, while still preserving the regret bounds?



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:

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



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:

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

- This means if we are dealing with a loss function $L : \mathcal{A} \times \mathcal{Z} \rightarrow \mathbb{R}$, 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: $L(a, z) - L(\tilde{a}, z) \leq (a - \tilde{a})^\top \nabla_a L(a, z), \quad \forall a, \tilde{a} \in \mathcal{A}, z \in \mathcal{Z}.$



ONLINE GRADIENT DESCENT: MOTIVATION

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

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

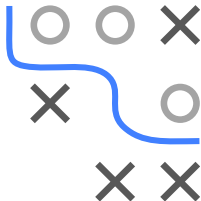


ONLINE GRADIENT DESCENT: MOTIVATION

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

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

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



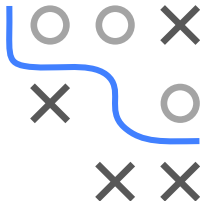
ONLINE GRADIENT DESCENT: MOTIVATION

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

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

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

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



ONLINE GRADIENT DESCENT: MOTIVATION

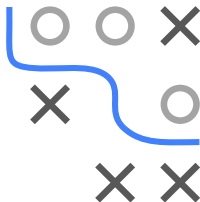
Reminder: $L(a, z) - L(\tilde{a}, z) \leq (a - \tilde{a})^\top \nabla_a L(a, z), \quad \forall a, \tilde{a} \in \mathcal{A}, z \in \mathcal{Z}.$

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

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

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

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



ONLINE GRADIENT DESCENT: MOTIVATION

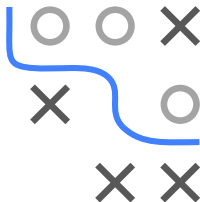
Reminder: $L(a, z) - L(\tilde{a}, z) \leq (a - \tilde{a})^\top \nabla_a L(a, z), \quad \forall a, \tilde{a} \in \mathcal{A}, z \in \mathcal{Z}.$

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

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

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

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

- 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,

$$\mathbf{a}_{t+1}^{\text{OGD}} = \mathbf{a}_t^{\text{OGD}} - \eta \nabla_{\mathbf{a}} L(\mathbf{a}_t^{\text{OGD}}, \mathbf{z}_t), \quad t = 1, \dots, T. \quad (1)$$

(Technical side note: For this update formula we assume that $\mathcal{A} = \mathbb{R}^d$. Moreover, the first action $\mathbf{a}_1^{\text{OGD}}$ is arbitrary.)

