Machine Learning Theory 2024
Lecture 10

Wouter M. Koolen

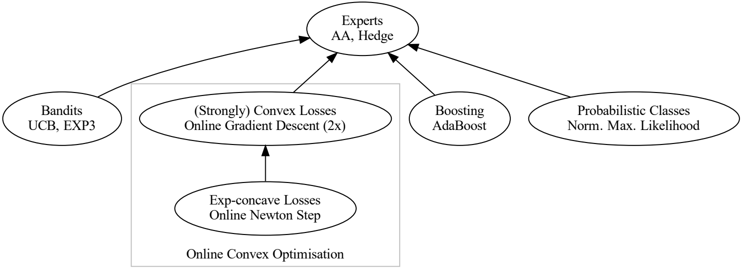
Download these slides now from elo.mastermath.nl!

Online Convex Optimisation

- Gradient Descent for Convex Losses
- Online to Batch Conversion
- Gradient Descent for Strongly Convex Losses

Recap

Overview of Second Half of Course



Material: course notes on MLT website.

Recap: Finite Classes

So far we have seen learning “finite sets”:
Our learning algorithms behave like the **best** among K strategies.

- K-Experts setting
  - Mix loss : Aggregating Algorithm
  - Dot loss : Hedge algorithm
- K-armed bandit settings
  - Adversarial bandit : EXP3
  - Stochastic bandit : UCB

Outlook: Beyond the Finite

What if we want to compete with **infinite** sets?

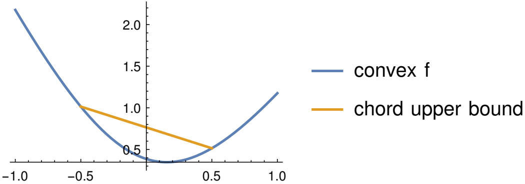
- Can we?
- How?

In each case, **lower bounds** grow with K: $\ln K$, $\sqrt{T \ln K}$, $\sqrt{TK \ln K}$, $K/\Delta \ln T$. So hopeless in the **unstructured** $K \to \infty$ case.

Today: compete with **continuous** sets of actions, parameterised such that the loss is a **convex** function of the action.

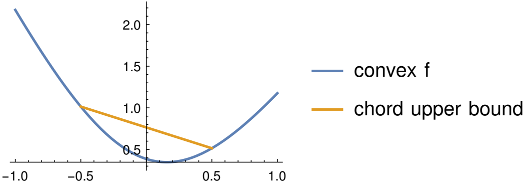
Convexity Review

Convex Functions I : definition



Fix a convex set $\mathcal{U} \subseteq \mathbb{R}^d$.

Convex Functions I : definition



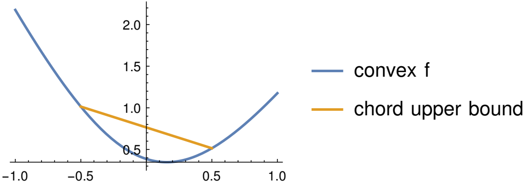
Fix a convex set $\mathcal{U} \subseteq \mathbb{R}^d$.

Definition

A function $f : \mathcal{U} \to \mathbb{R}$ is convex if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{U}$ and weights $\theta \in [0, 1]$,

$$f(\theta \boldsymbol{x} + (1 - \theta)\boldsymbol{y}) \;\leq\; \theta f(\boldsymbol{x}) + (1 - \theta) f(\boldsymbol{y}).$$

Convex Functions I : definition



Fix a convex set $\mathcal{U} \subseteq \mathbb{R}^d$.

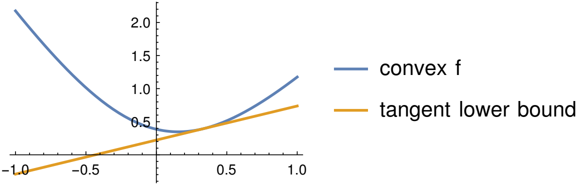
Definition

A function $f : \mathcal{U} \to \mathbb{R}$ is convex if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{U}$ and weights $\theta \in [0, 1]$,

$$f(\theta \boldsymbol{x} + (1 - \theta)\boldsymbol{y}) \;\leq\; \theta f(\boldsymbol{x}) + (1 - \theta) f(\boldsymbol{y}).$$

Extends to arbitrary mixtures: $f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$ (Jensen).

Convex Functions II : tangent bound

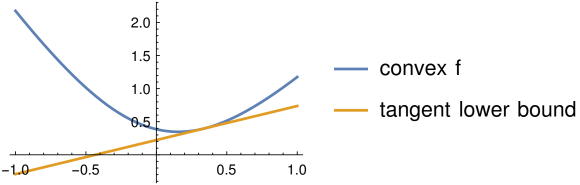


Fact

A differentiable function $f : \mathcal{U} \to \mathbb{R}$ *is convex iff for all* $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{U}$

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) \ \geq \ \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f(\boldsymbol{x}) \rangle$$

Convex Functions II : tangent bound



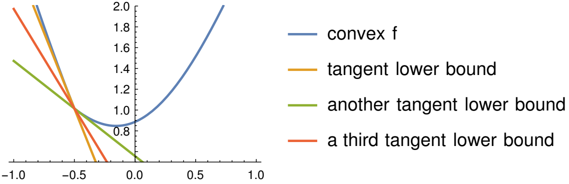
Fact

A differentiable function $f : \mathcal{U} \to \mathbb{R}$ is convex iff for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{U}$

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) \ \geq \ \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f(\boldsymbol{x}) \rangle$$

Symmetrically, $\langle \boldsymbol{y} - \boldsymbol{x}, \nabla f(\boldsymbol{y}) \rangle \ \geq \ f(\boldsymbol{y}) - f(\boldsymbol{x})$.

Convex Functions III : sub-gradient



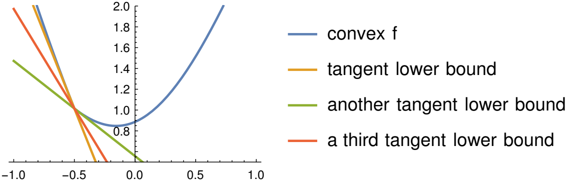
Fact (Sub-gradient)

For any convex $f : \mathcal{U} \to \mathbb{R}$, possibly non-differentiable, and point $\boldsymbol{x} \in \mathcal{U}$, there always exists **some** *vector $\boldsymbol{g} \in \mathbb{R}^d$ such that for all $\boldsymbol{y} \in \mathcal{U}$*

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) \geq \langle \boldsymbol{y} - \boldsymbol{x}, \boldsymbol{g} \rangle$$

Any such vector $\boldsymbol{g}$ is called a **sub-gradient** *(of f at $\boldsymbol{x}$).*

Convex Functions III : sub-gradient



Fact (Sub-gradient)

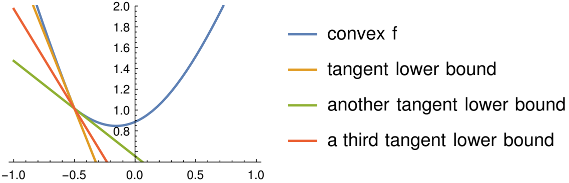
For any convex $f : \mathcal{U} \to \mathbb{R}$, possibly non-differentiable, and point $\boldsymbol{x} \in \mathcal{U}$, there always exists **some** *vector $\boldsymbol{g} \in \mathbb{R}^d$ such that for all $\boldsymbol{y} \in \mathcal{U}$*

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) \geq \langle \boldsymbol{y} - \boldsymbol{x}, \boldsymbol{g} \rangle$$

Any such vector $\boldsymbol{g}$ is called a **sub-gradient** *(of f at $\boldsymbol{x}$).*

The gradient of a differentiable function is a sub-gradient.

Convex Functions III : sub-gradient



Fact (Sub-gradient)

For any convex $f : \mathcal{U} \to \mathbb{R}$, possibly non-differentiable, and point $\boldsymbol{x} \in \mathcal{U}$, there always exists **some** *vector $\boldsymbol{g} \in \mathbb{R}^d$ such that for all $\boldsymbol{y} \in \mathcal{U}$*

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) \geq \langle \boldsymbol{y} - \boldsymbol{x}, \boldsymbol{g} \rangle$$

Any such vector $\boldsymbol{g}$ is called a **sub-gradient** *(of f at $\boldsymbol{x}$).*

The gradient of a differentiable function is a sub-gradient.

We will abuse notation and denote **any** sub-gradient by $\nabla f(\boldsymbol{x})$.

Online Convex Optimisation

Online Convex Optimisation

General yet simple sequential decision problem.

Fix a convex set $\mathcal{U} \subseteq \mathbb{R}^d$.

Protocol

For $t = 1, 2, \dots$

- Learner chooses a point $\boldsymbol{w}_t \in \mathcal{U}$.
- Adversary reveals convex loss function $f_t : \mathcal{U} \to \mathbb{R}$.
- Learner's loss is $f_t(\boldsymbol{w}_t)$

Online Convex Optimisation

General yet simple sequential decision problem.

Fix a convex set $\mathcal{U} \subseteq \mathbb{R}^d$.

Protocol

For $t = 1, 2, \ldots$

- Learner chooses a point $\boldsymbol{w}_t \in \mathcal{U}$.
- Adversary reveals convex loss function $f_t : \mathcal{U} \to \mathbb{R}$.
- Learner's loss is $f_t(\boldsymbol{w}_t)$

Objective:

Regret w.r.t. best point after T rounds:

$$R_T \;=\; \max_{\boldsymbol{u} \in \mathcal{U}} \sum_{t=1}^{T} \left(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u}) \right)$$

Example loss functions

Setting	loss function $f_t(\boldsymbol{u})$
Hedge setting	$\boldsymbol{u}^\intercal \boldsymbol{\ell}_t$
Point prediction	$\left\|\boldsymbol{u} - \boldsymbol{x}_t\right\|^2$
Regression	$(\boldsymbol{u}^\intercal \boldsymbol{x}_t - y_t)^2$
Logistic regression	$\ln\left(1 + e^{-y_t \boldsymbol{u}^\intercal \boldsymbol{x}_t}\right)$
Hinge loss	$\max\{0, 1 - y_t \boldsymbol{u}^\intercal \boldsymbol{x}_t\}$
Investment	$-\ln(\boldsymbol{u}^\intercal \boldsymbol{x}_t)$
Offline optimisation	$f(\boldsymbol{u})$

Online Gradient Descent (OGD)

Let $\mathcal{U}$ be a closed convex set containing $\mathbf{0}$.

Definition

Online Gradient Descent with learning rate $\eta > 0$ plays

$$\boldsymbol{w}_1 = \mathbf{0} \qquad \text{and} \qquad \boldsymbol{w}_{t+1} \;=\; \Pi_{\mathcal{U}}\left(\boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t)\right)$$

where $\Pi_{\mathcal{U}}(\boldsymbol{w}) = \arg\min_{\boldsymbol{u} \in \mathcal{U}} \|\boldsymbol{u} - \boldsymbol{w}\|$ is the projection onto $\mathcal{U}$.

Online Gradient Descent (OGD)

Theorem

Let $\|\nabla f_t(\boldsymbol{u})\| \leq G$ *and* $\|\boldsymbol{u}\| \leq D$ *for all* $\boldsymbol{u} \in \mathcal{U}$*. Then*

$$R_T \;=\; \max_{\boldsymbol{u}\in\mathcal{U}} \sum_{t=1}^{T} (f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u})) \;\leq\; \frac{1}{2\eta} D^2 + \frac{\eta}{2} T G^2$$

Online Gradient Descent (OGD)

Theorem

Let $\|\nabla f_t(\boldsymbol{u})\| \leq G$ and $\|\boldsymbol{u}\| \leq D$ for all $\boldsymbol{u} \in \mathcal{U}$. Then

$$R_T \;=\; \max_{\boldsymbol{u}\in\mathcal{U}} \sum_{t=1}^{T} \left(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u})\right) \;\leq\; \frac{1}{2\eta} D^2 + \frac{\eta}{2} T G^2$$

Corollary

Tuning $\eta = \frac{D}{G\sqrt{T}}$ results in

$$R_T \;\leq\; DG\sqrt{T}$$

Pythagorean Inequality

Lemma (Pythagorean Inequality)

Fix a closed convex set $\mathcal{U} \subseteq \mathbb{R}^d$. *Let* $\boldsymbol{x} \in \mathcal{U}, \boldsymbol{y} \in \mathbb{R}^d$ *and*

$$\hat{\boldsymbol{y}} \;=\; \Pi_{\mathcal{U}}(\boldsymbol{y}) \;=\; \arg\min_{\boldsymbol{u} \in \mathcal{U}} \|\boldsymbol{u} - \boldsymbol{y}\|^2.$$

Then

$$\|\boldsymbol{x} - \hat{\boldsymbol{y}}\|^2 + \|\hat{\boldsymbol{y}} - \boldsymbol{y}\|^2 \;\leq\; \|\boldsymbol{x} - \boldsymbol{y}\|^2$$

NB: not to be confused with the **triangle inequality**
$\|\boldsymbol{x} - \boldsymbol{y}\| \leq \|\boldsymbol{x} - \hat{\boldsymbol{y}}\| + \|\hat{\boldsymbol{y}} - \boldsymbol{y}\|$.

Proof of GD regret bound I

Fix any $\boldsymbol{u} \in \mathcal{U}$. We have

$$f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u}) \;\leq\; \langle \boldsymbol{w}_t - \boldsymbol{u}, \nabla f_t(w_t) \rangle$$

Moreover,

$$\begin{aligned} \|\boldsymbol{w}_{t+1} - \boldsymbol{u}\|^2 \;&=\; \|\Pi_{\mathcal{U}}\left(\boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t)\right) - \boldsymbol{u}\|^2 \\ &\overset{\text{Pyth.Ineq.}}{\leq} \|\boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t) - \boldsymbol{u}\|^2 \\ &=\; \|\boldsymbol{w}_t - \boldsymbol{u}\|^2 - 2\eta \langle \boldsymbol{w}_t - \boldsymbol{u}, \nabla f_t(\boldsymbol{w}_t) \rangle + \eta^2 \|\nabla f_t(\boldsymbol{w}_t)\|^2 \end{aligned}$$

Hence

$$\langle \boldsymbol{w}_t - \boldsymbol{u}, \nabla f_t(\boldsymbol{w}_t) \rangle \;\leq\; \frac{\|\boldsymbol{w}_t - \boldsymbol{u}\|^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{u}\|^2}{2\eta} + \frac{\eta}{2} \|\nabla f_t(\boldsymbol{w}_t)\|^2$$

Proof of GD regret bound II

Summing over T rounds, we find

$$\begin{aligned}
\sum_{t=1}^{T}(f_t(\boldsymbol{w}_t)-f_t(\boldsymbol{u})) &\leq \sum_{t=1}^{T}\langle \boldsymbol{w}_t-\boldsymbol{u}, \nabla f_t(\boldsymbol{w}_t)\rangle \\
&\leq \underbrace{\sum_{t=1}^{T}\frac{\|\boldsymbol{w}_t-\boldsymbol{u}\|^2-\|\boldsymbol{w}_{t+1}-\boldsymbol{u}\|^2}{2\eta}}_{\text{telescopes}}+\frac{\eta}{2}\sum_{t=1}^{T}\|\nabla f_t(\boldsymbol{w}_t)\|^2 \\
&\leq \frac{\|\boldsymbol{u}\|^2-\cancel{\|\boldsymbol{w}_{T+1}-\boldsymbol{u}\|^2}}{2\eta}+\frac{\eta}{2}\sum_{t=1}^{T}\|\nabla f_t(\boldsymbol{w}_t)\|^2 \\
&\leq \frac{D^2}{2\eta}+\frac{\eta}{2}TG^2
\end{aligned}$$

Online to Batch Conversion

Online to Batch Conversion

Goal: obtain an estimator $\hat{\boldsymbol{w}}_T$ with small expected excess risk.

$$\mathop{\mathbb{E}}_{f_1,\ldots,f_T}\left[\mathop{\mathbb{E}}_{f}\left[f(\hat{\boldsymbol{w}}_T) - f(\boldsymbol{u}^*)\right]\right] \;\leq\; \text{small}$$

where the training set $f_1, \ldots, f_T$ and the test sample f are drawn i.i.d. and $\boldsymbol{u}^*$ optimises the risk $\boldsymbol{u} \mapsto \mathbb{E}_f[f(\boldsymbol{u})]$.

Online to Batch Conversion

Goal: obtain an estimator $\hat{\boldsymbol{w}}_T$ with small expected excess risk.

$$\mathop{\mathbb{E}}_{f_1,\ldots,f_T}\left[\mathop{\mathbb{E}}_{f}\left[f(\hat{\boldsymbol{w}}_T) - f(\boldsymbol{u}^*)\right]\right] \;\le\; \text{small}$$

where the training set $f_1, \ldots, f_T$ and the test sample f are drawn i.i.d. and $\boldsymbol{u}^*$ optimises the risk $\boldsymbol{u} \mapsto \mathbb{E}_f[f(\boldsymbol{u})]$.

Idea: use online learning algorithm. Given training sample $f_1, \ldots, f_T$, the algorithm picks $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_T$. Let us define the *average iterate estimator*

$$\hat{\boldsymbol{w}}_T \;=\; \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{w}_t.$$

Online to Batch Conversion

Goal: obtain an estimator $\hat{\boldsymbol{w}}_T$ with small expected excess risk.

$$\mathop{\mathbb{E}}_{f_1,\ldots,f_T}\left[\mathop{\mathbb{E}}_{f}\left[f(\hat{\boldsymbol{w}}_T) - f(\boldsymbol{u}^*)\right]\right] \;\le\; \text{small}$$

where the training set $f_1, \ldots, f_T$ and the test sample f are drawn i.i.d. and $\boldsymbol{u}^*$ optimises the risk $\boldsymbol{u} \mapsto \mathbb{E}_f[f(\boldsymbol{u})]$.

Idea: use online learning algorithm. Given training sample $f_1, \ldots, f_T$, the algorithm picks $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_T$. Let us define the *average iterate estimator*

$$\hat{\boldsymbol{w}}_T \;=\; \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{w}_t.$$

Theorem

An online regret bound $R_T \le B(T)$ implies

$$\mathop{\mathbb{E}}_{\textit{iid } f_1,\,\ldots,\,f_T,\,f}\left[f\left(\hat{\boldsymbol{w}}_T\right) - f(\boldsymbol{u}^*)\right] \;\le\; \frac{B(T)}{T}$$

Online to Batch Proof

$$\begin{aligned}
&\underset{\text{iid } f_1, \ldots, f_T, f}{\mathbb{E}} \left[f\left(\hat{\boldsymbol{w}}_T\right) - f(\boldsymbol{u}^*) \right] \\
&\leq \underset{\text{iid } f_1, \ldots, f_T, f}{\mathbb{E}} \left[\frac{1}{T} \sum_{t=1}^{T} \left(f(\boldsymbol{w}_t) - f(\boldsymbol{u}^*) \right) \right] \\
&= \underset{\text{iid } f_1, \ldots, f_T, f}{\mathbb{E}} \left[\frac{1}{T} \sum_{t=1}^{T} \left(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u}^*) \right) \right] \leq \frac{B(T)}{T}
\end{aligned}$$

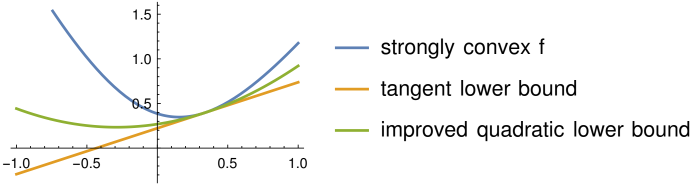
The first step is convexity of f. The last step uses that f and f_t have the same distribution (and $\boldsymbol{w}_t$ is not a function of f_t).

Online Strongly Convex Optimisation

Structure

What if I **know more** about my setting than **convexity of the loss function**? Can I learn faster?

Strongly Convex Case

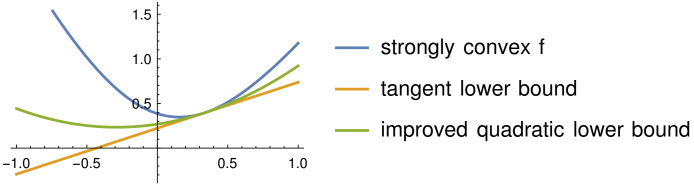


Definition

A function $f : \mathcal{U} \to \mathbb{R}$ is *strongly convex* to degree $\alpha \geq 0$ if

$$f(\boldsymbol{u}) - f(\boldsymbol{w}) \;\geq\; \langle \boldsymbol{u} - \boldsymbol{w}, \nabla f(\boldsymbol{w}) \rangle + \frac{\alpha}{2} \|\boldsymbol{u} - \boldsymbol{w}\|^2$$

Strongly Convex Case



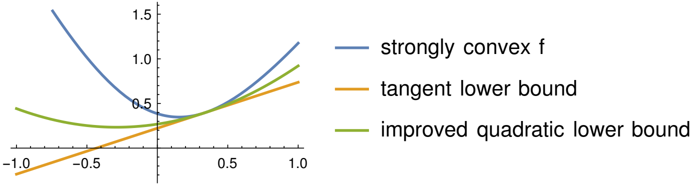
Definition

A function $f : \mathcal{U} \to \mathbb{R}$ is *strongly convex* to degree $\alpha \geq 0$ if

$$f(\boldsymbol{u}) - f(\boldsymbol{w}) \;\geq\; \langle \boldsymbol{u} - \boldsymbol{w}, \nabla f(\boldsymbol{w}) \rangle + \frac{\alpha}{2} \|\boldsymbol{u} - \boldsymbol{w}\|^2$$

Example: $f(\boldsymbol{w}) = \|\boldsymbol{w} - \boldsymbol{x}_t\|^2$.

Strongly Convex Case



Definition

A function $f : \mathcal{U} \to \mathbb{R}$ is *strongly convex* to degree $\alpha \geq 0$ if

$$f(\boldsymbol{u}) - f(\boldsymbol{w}) \ \geq \ \langle \boldsymbol{u} - \boldsymbol{w}, \nabla f(\boldsymbol{w}) \rangle + \frac{\alpha}{2} \|\boldsymbol{u} - \boldsymbol{w}\|^2$$

Example: $f(\boldsymbol{w}) = \|\boldsymbol{w} - \boldsymbol{x}_t\|^2$.

Idea: could this extra knowledge help in the regret rate?

Online Gradient Descent
with time-varying learning rate

Definition (OGD with time-varying learning rate)

$$\boldsymbol{w}_1 \;=\; \mathbf{0} \qquad \text{and} \qquad \boldsymbol{w}_{t+1} \;=\; \Pi_{\mathcal{U}}\left(\boldsymbol{w}_t - \eta_t \nabla f_t(\boldsymbol{w}_t)\right)$$

Online Gradient Descent with time-varying learning rate

Definition (OGD with time-varying learning rate)

$$\boldsymbol{w}_1 \;=\; \mathbf{0} \qquad \text{and} \qquad \boldsymbol{w}_{t+1} \;=\; \Pi_{\mathcal{U}}\left(\boldsymbol{w}_t - \eta_t \nabla f_t(\boldsymbol{w}_t)\right)$$

Theorem

For α-strongly convex loss functions, OGD with learning rate $\eta_t = \frac{1}{\alpha t}$ ensures

$$R_T \;\leq\; \frac{G^2}{2\alpha}\left(1 + \ln T\right).$$

Proof I

Exactly as for the convex case, the update rule ensures

$$\begin{aligned}
\|\boldsymbol{w}_{t+1}-\boldsymbol{u}\|^2 &= \|\Pi_{\mathcal{U}}(\boldsymbol{w}_t-\eta_t\nabla f_t(\boldsymbol{w}_t))-\boldsymbol{u}\|^2 \\
&\overset{\text{Pyth.Ineq.}}{\leq} \|\boldsymbol{w}_t-\eta_t\nabla f_t(\boldsymbol{w}_t)-\boldsymbol{u}\|^2 \\
&= \|\boldsymbol{w}_t-\boldsymbol{u}\|^2 - 2\eta_t\langle \boldsymbol{w}_t-\boldsymbol{u}, \nabla f_t(\boldsymbol{w}_t)\rangle + \eta_t^2\|\nabla f_t(\boldsymbol{w}_t)\|^2
\end{aligned}$$

Combination with strong convexity gives

$$\begin{aligned}
& f_t(\boldsymbol{w}_t)-f_t(\boldsymbol{u}) \\
&\quad \leq \langle \boldsymbol{w}_t-\boldsymbol{u}, \nabla f_t(\boldsymbol{w}_t)\rangle - \frac{\alpha}{2}\|\boldsymbol{w}_t-\boldsymbol{u}\|^2 \\
&\quad \leq \frac{\|\boldsymbol{w}_t-\boldsymbol{u}\|^2-\|\boldsymbol{w}_{t+1}-\boldsymbol{u}\|^2+\eta_t^2\|\nabla f_t(\boldsymbol{w}_t)\|^2}{2\eta_t} - \frac{\alpha}{2}\|\boldsymbol{w}_t-\boldsymbol{u}\|^2 \\
&\quad = \|\boldsymbol{w}_t-\boldsymbol{u}\|^2\left(\frac{1}{2\eta_t}-\frac{\alpha}{2}\right) - \frac{\|\boldsymbol{w}_{t+1}-\boldsymbol{u}\|^2}{2\eta_t} + \frac{\eta_t\|\nabla f_t(\boldsymbol{w}_t)\|^2}{2}
\end{aligned}$$

Proof II

Summing over rounds gives

$$
\begin{aligned}
&\sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u}) \\
&\leq \sum_{t=1}^{T} \left(\|\boldsymbol{w}_t - \boldsymbol{u}\|^2 \left(\frac{1}{2\eta_t} - \frac{\alpha}{2} \right) - \frac{\|\boldsymbol{w}_{t+1} - \boldsymbol{u}\|^2}{2\eta_t} + \frac{\eta_t \|\nabla f_t(\boldsymbol{w}_t)\|^2}{2} \right) \\
&= \|\boldsymbol{w}_1 - \boldsymbol{u}\|^2 \left(\frac{1}{2\eta_1} - \frac{\alpha}{2} \right) + \sum_{t=2}^{T} \|\boldsymbol{w}_t - \boldsymbol{u}\|^2 \left(\frac{1}{2\eta_t} - \frac{\alpha}{2} - \frac{1}{2\eta_{t-1}} \right) \\
&\quad - \frac{\|\boldsymbol{w}_{T+1} - \boldsymbol{u}\|^2}{2\eta_T} + \sum_{t=1}^{T} \frac{\eta_t \|\nabla f_t(\boldsymbol{w}_t)\|^2}{2}
\end{aligned}
$$

Key idea for telescoping is to cancel coefficient on $\|\boldsymbol{w}_t - \boldsymbol{u}\|^2$ in the sum:

$$
\frac{1}{2\eta_t} - \frac{\alpha}{2} - \frac{1}{2\eta_{t-1}} = 0
$$

Proof III

This yields recurrence

$$\eta_t \;=\; \frac{1}{\frac{1}{\eta_{t-1}} + \alpha}$$

Cancelling the coefficient on $\|\boldsymbol{w}_1 - \boldsymbol{u}\|^2$ gives starting point $\eta_1 \;=\; \frac{1}{\alpha}$. This leads to overall solution $\eta_t = \frac{1}{\alpha t}$. Plugging that in, we find

$$\sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u}) \;\leq\; \sum_{t=1}^{T} \frac{\|\nabla f_t(\boldsymbol{w}_t)\|^2}{2\alpha t} \;\leq\; \frac{G^2}{2\alpha}(1 + \ln T)$$

Conclusion

Tools for learning in convex settings.

- Guaranteed robustness against adversarial losses
- Efficient
- Building block for
  - Learning in non-convex settings (AdaGrad for DNN)
  - Learning in games
  - Non-convex games (GANs)
  - . . .