MetaGrad: Faster Convergence **Without Curvature** in Online Convex Optimization

Wouter M. Koolen Tim van Erven

CWI
Centrum Wiskunde & Informatica

Universiteit Leiden

Inria Lille
Friday 15th April, 2016

This talk

- Online Convex Optimization
- Learning the Learning rate
- Second-order (variance) bounds (individual sequence)
- Fast rates without curvature

Outline

Fundamental Learning Model: Online Convex Optimization

- In round $t = 1, 2, \ldots$
  - Learner predicts $\boldsymbol{w}_t$ (from unit ball)
  - Encounter convex loss function $f_t(\boldsymbol{u}) : \mathbb{R}^d \to \mathbb{R}$

  - Learner
    - observes gradient $\boldsymbol{g}_t := \nabla f_t(\boldsymbol{w}_t)$ (from unit ball)
    - incurs loss $f_t(\boldsymbol{w}_t)$

Fundamental Learning Model: Online Convex Optimization

- In round $t = 1, 2, \ldots$
  - Learner predicts $\boldsymbol{w}_t$ (from unit ball)
  - Encounter convex loss function $f_t(\boldsymbol{u}) : \mathbb{R}^d \to \mathbb{R}$


  - Learner
    - observes gradient $\boldsymbol{g}_t \coloneqq \nabla f_t(\boldsymbol{w}_t)$ (from unit ball)
    - incurs loss $f_t(\boldsymbol{w}_t)$

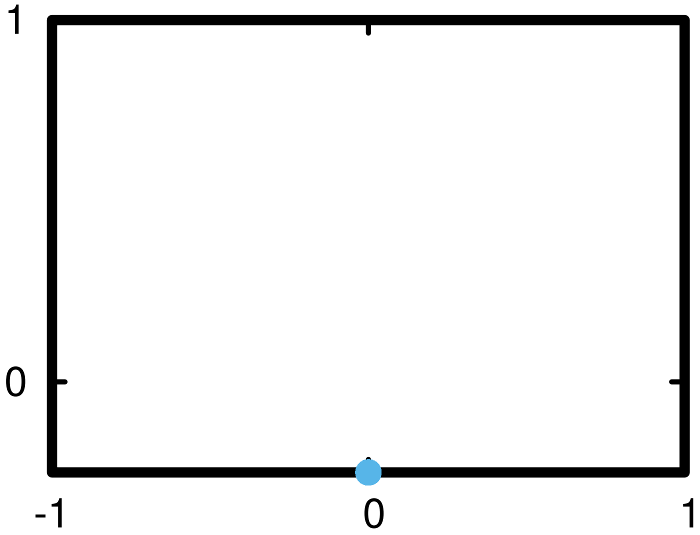
- The goal is to have small **regret**

$$R_T^{\boldsymbol{u}} \;\coloneqq\; \underbrace{\sum_{t=1}^{T} f_t(\boldsymbol{w}_t)}_{\text{Learner}} - \underbrace{\sum_{t=1}^{T} f_t(\boldsymbol{u})}_{\text{Point } \boldsymbol{u}}$$

with respect to **every** point $\boldsymbol{u}$.

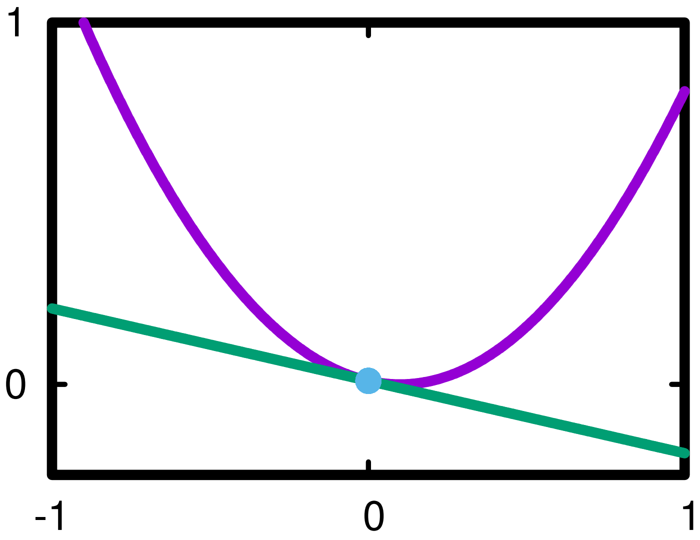
The Learner's Perspective

Round 1: Learner plays $\boldsymbol{w}_1 = \mathbf{0}$



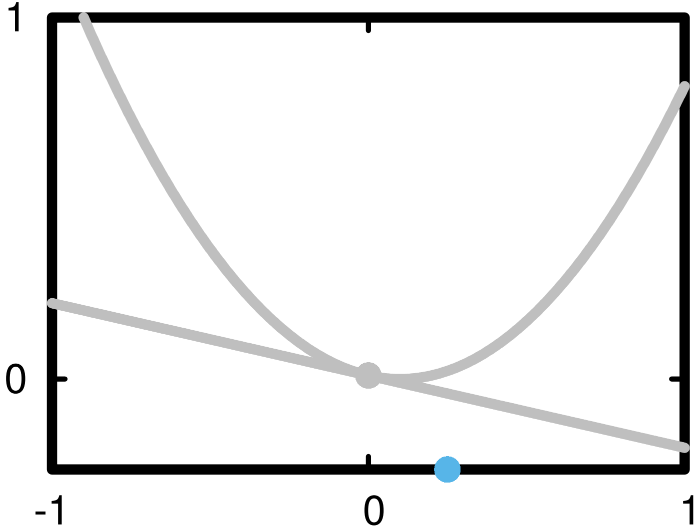
The Learner's Perspective

Round 1: Learner incurs $f_1(\boldsymbol{w}_1)$ and sees $\boldsymbol{g}_1 = \nabla f_1(\boldsymbol{w}_1)$



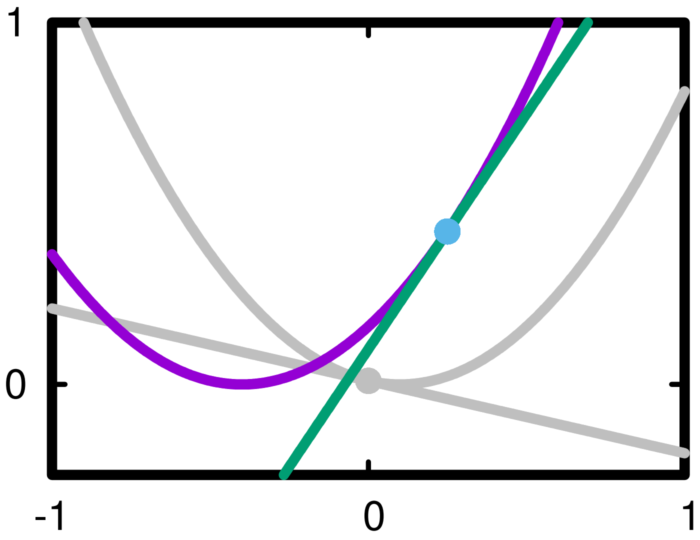
The Learner's Perspective

Round 2: Learner plays $w_1 = 1/4$



The Learner's Perspective

Round 2: Learner incurs $f_2(\boldsymbol{w}_2)$ and sees $\boldsymbol{g}_2 = \nabla f_2(\boldsymbol{w}_2)$

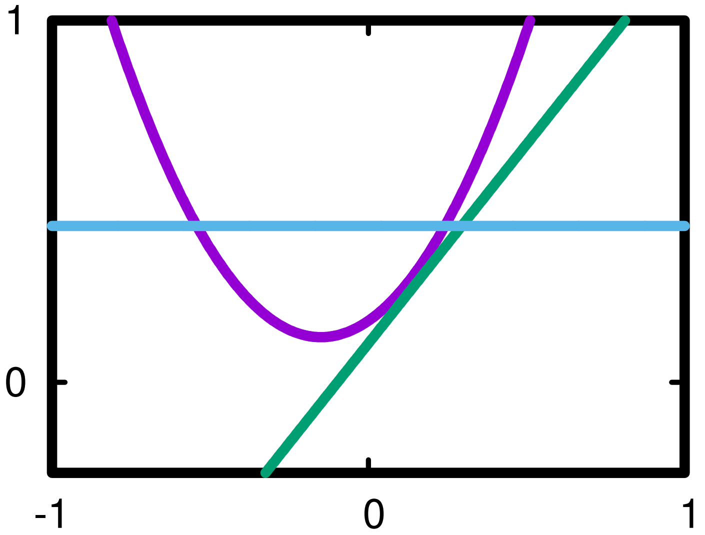


The Learner's Perspective

...

The Learner's Perspective

Evaluate Learner using **regret**: $R_T^{\boldsymbol{u}} = \sum_{t=1}^{T} \left(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u})\right)$



State of the Art

Online gradient descent

$$\boldsymbol{w}_{t+1} \;=\; \boldsymbol{w}_t - {\color{red}\eta}\boldsymbol{g}_t$$

recall $\boldsymbol{g}_t = \nabla f_t(\boldsymbol{w}_t)$

State of the Art

Online gradient descent

$$\boldsymbol{w}_{t+1} \;=\; \boldsymbol{w}_t - \eta \boldsymbol{g}_t$$

recall $\boldsymbol{g}_t = \nabla f_t(\boldsymbol{w}_t)$

OGD bound: After T rounds,

$$R_T^{\boldsymbol{u}} \;\leq\; O\left(\sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|^2}\right) \qquad \text{for all } \boldsymbol{u} \text{ with } \|\boldsymbol{u}\| \leq 1.$$

Bounds Reveal Our Dearest Hopes

Always have **worst-case guarantee**

$$R_T^{\boldsymbol{u}} \leq O\left(\sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|^2}\right) \leq O(\sqrt{T}).$$

Yet bound says we **might get lucky**

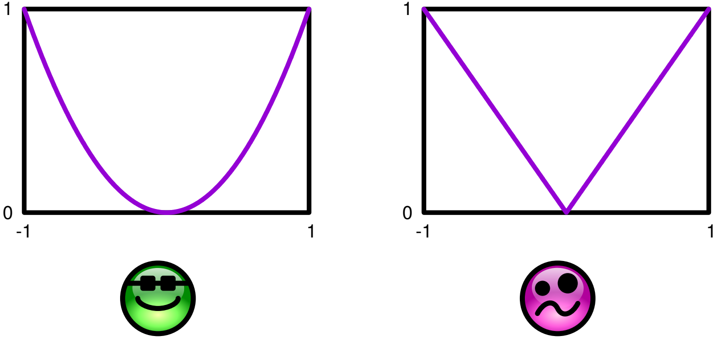
For smooth functions f_t with common optimum $\boldsymbol{u}^*$, as $\boldsymbol{w}_t \to \boldsymbol{u}^*$, we have $\boldsymbol{g}_t \to \boldsymbol{0}$, and

$$\sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|^2} \ll \sqrt{T}$$

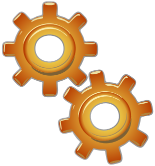
grows **much slower** than $\sqrt{T}$.

What We Hope Happens

$$R_T^{\boldsymbol{u}} \leq O\left(\sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|^2}\right)$$



Can We Do Better?



No **in general**: matching lower bound.

$$R_T^{\boldsymbol{u}} \;\geq\; \Omega(\sqrt{T})$$

Yes, **with curvature**:

$$R_T^{\boldsymbol{u}} \;\leq\; O(\ln T)$$

- Strongly convex: $\boldsymbol{I} \preceq \nabla^2 f(\boldsymbol{u})$, e.g.

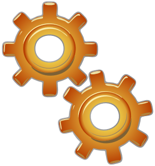
  $$f_t(\boldsymbol{u}) \;=\; \|\boldsymbol{u} - \boldsymbol{y}_t\|^2$$

  $\Rightarrow$ gradient descent with small η
- Exp-concave: $\nabla f(\boldsymbol{u}) \nabla f(\boldsymbol{u})^\intercal \preceq \nabla^2 f(\boldsymbol{u})$, e.g.

  $$f_t(\boldsymbol{u}) \;=\; -\ln\left(1 + \boldsymbol{y}_t^\intercal \boldsymbol{u}\right)$$

  $\Rightarrow$ Online Newton Step

Can We Do Better?



But do we really need **curvature**?

This talk: no, **stability** is enough.

New algorithm MetaGrad:
Separate learning rate η for each point $\boldsymbol{u}$

Outline

Online Convex Optimization

A New Type of Guarantee

Fast Rates

MetaGrad Algorithm

Refined Bound

Recall bound for gradient descent:

$$R_T^{\boldsymbol{u}} \;\leq\; O\left(\sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|^2}\right)$$

New bound for MetaGrad:

$$R_T^{\boldsymbol{u}} \;\leq\; O\left(\sqrt{V_T^{\boldsymbol{u}}\, d \ln T}\right) \quad \text{where} \quad V_T^{\boldsymbol{u}} \;:=\; \sum_{t=1}^{T} \left((\boldsymbol{w}_t - \boldsymbol{u})^\intercal \boldsymbol{g}_t\right)^2$$

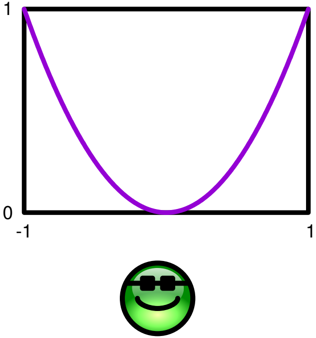
Data-dependent. Whoa! Ouroboric.

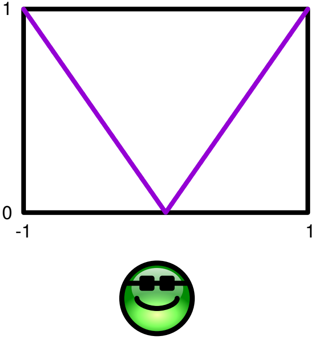
Always improvement:

$$\left((\boldsymbol{w}_t - \boldsymbol{u})^\intercal \boldsymbol{g}_t\right)^2 \;\leq\; \|\boldsymbol{w}_t - \boldsymbol{u}\|^2 \|\boldsymbol{g}_t\|^2$$

Now What We Hope Happens

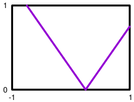
$$R_T^{\boldsymbol{u}} \leq O\left(\sqrt{\sum_{t=1}^{T}\left((\boldsymbol{w}_t - \boldsymbol{u})^\intercal \boldsymbol{g}_t\right)^2}\right)$$





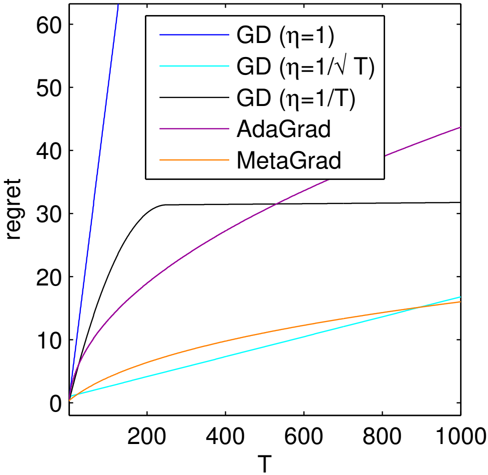
Outline

Does it Really Work?



Offline optimization (fixed function):

$$f_t(u) \; = \; |u - 1/4|$$



The “Fast Rates” Pipeline

Combine

- refined individual-sequence regret bound

$$R_T^{\boldsymbol{u}} \;\le\; \sqrt{V_T^{\boldsymbol{u}} d \ln T} \quad \forall \boldsymbol{u}$$

- Special-purpose argument that for best $\boldsymbol{u}^*$

$$\boxed{V_T^{\boldsymbol{u}^*} \;\le\; R_T^{\boldsymbol{u}^*}}$$

- Profit!

$$R_T^{\boldsymbol{u}^*} \;\le\; \sqrt{R_T^{\boldsymbol{u}^*} d \ln T} \qquad \text{so} \qquad R_T^{\boldsymbol{u}^*} \;\le\; d \ln T$$

Significant Improvement: Fixed Function

Any fixed $f_t(\boldsymbol{u}) = f(\boldsymbol{u})$.
Let $\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}} f(\boldsymbol{u})$ be the offline minimiser.

Crux: $(\boldsymbol{w}_t - \boldsymbol{u}^*)^\intercal \boldsymbol{g}_t \in [0, 2]$.

Now from the regret bound

$$R_T^{\boldsymbol{u}^*} \;\le\; \sum_{t=1}^T (\boldsymbol{w}_t - \boldsymbol{u}^*)^\intercal \boldsymbol{g}_t \;\le\; \sqrt{V_T^{\boldsymbol{u}^*} d \ln T}$$

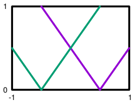
and **special-purpose** observation

$$\boxed{V_T^{\boldsymbol{u}^*} \;=\; \sum_{t=1}^T \left((\boldsymbol{u}^* - \boldsymbol{w}_t)^\intercal \boldsymbol{g}_t\right)^2 \;\le\; 2 \sum_{t=1}^T (\boldsymbol{w}_t - \boldsymbol{u}^*)^\intercal \boldsymbol{g}_t}$$

we can solve for $V_T^{\boldsymbol{u}^*}$ to find $V_T^{\boldsymbol{u}^*} \le d \ln T$ and hence

$$R_T^{\boldsymbol{u}^*} \;\le\; \sqrt{2} d \ln T$$

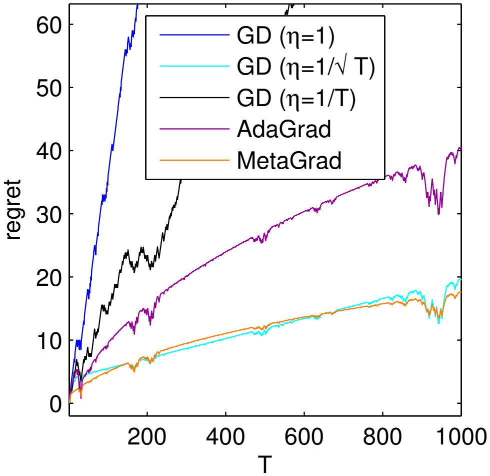
Does It Really Actually Work?



Stochastic optimization:

$$f_t(u) \; = \; |u - x_t|$$

where $x_t = \pm\frac{1}{2}$ i.i.d. with probability 0.4 and 0.6.

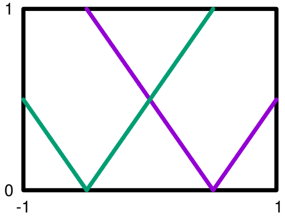


What's Going On, Really?

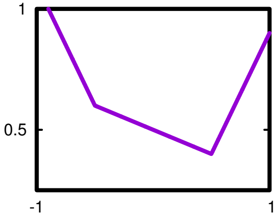
Stochastic optimization:

$$f_t(u) \;=\; |u - x_t|$$

where $x_t = \pm\frac{1}{2}$ i.i.d. with probability 0.4 and 0.6.



Individual functions



Long-term average

Stable minimum easy to converge to

Significant Improvement: Stochastic Case

Consider i.i.d.

$$f \sim \mathbb{P} \qquad \text{with} \qquad \boldsymbol{u}^* = \arg\min_{\boldsymbol{u}} \mathbb{E}[f(\boldsymbol{u})]$$

Condition: there is a $c > 0$ such that

$$\forall \boldsymbol{w} : (\boldsymbol{w} - \boldsymbol{u}^*)^\intercal \, \mathbb{E}\left[\nabla f(\boldsymbol{w}) \nabla f(\boldsymbol{w})^\intercal\right] (\boldsymbol{w} - \boldsymbol{u}^*) \;\leq\; c(\boldsymbol{w} - \boldsymbol{u}^*)^\intercal \, \mathbb{E}\left[\nabla f(\boldsymbol{w})\right]$$

Now from the **special-case** condition

$$\mathbb{E}[V_T^{\boldsymbol{u}^*}] \;\leq\; c\,\mathbb{E}\left[\sum_{t=1}^T (\boldsymbol{w}_t - \boldsymbol{u}^*)^\intercal \nabla f(\boldsymbol{w}_t)\right]$$

and by the **generic** regret bound, in expectation,

$$\mathbb{E}[R_T^{\boldsymbol{u}^*}] \leq \mathbb{E}\left[\sum_{t=1}^T (\boldsymbol{w}_t - \boldsymbol{u}^*)^\intercal \nabla f(\boldsymbol{w}_t)\right] \leq \mathbb{E}\left[\sqrt{V_T^{\boldsymbol{u}^*} d \ln T}\right]$$

and by Jensen's inequality $\mathbb{E}\left[\sqrt{V_T^{\boldsymbol{u}^*}}\right] \leq \sqrt{\mathbb{E}\left[V_T^{\boldsymbol{u}^*}\right]}$, so that

$$\mathbb{E}[R_T^{\boldsymbol{u}^*}] \;\leq\; \sqrt{c} d \ln T$$

Outline

Online Convex Optimization

A New Type of Guarantee

Fast Rates

MetaGrad Algorithm

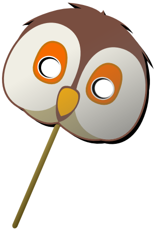
Our Approach in a Nutshell

1. Replace actual loss $f_t(\boldsymbol{u})$ by **surrogate loss** $\ell_t^{\eta}(\boldsymbol{u})$
   - parametrised by learning rate η
   - exp-concave in $\boldsymbol{u}$
   - So can get good bound for surrogate regret
2. Exponentially spaced **grid** $\eta_1, \eta_2, \ldots, \eta_{\log(T)}$ $(\eta_i = 2^{-i})$.
3. Off-the-shelf exp-concave **Slave** for grid point η_i predicts

$$\boldsymbol{w}_1^{\eta_i}, \boldsymbol{w}_2^{\eta_i}, \ldots$$

4. At each round t, **Master** aggregates $\boldsymbol{w}_t^{\eta_1}, \boldsymbol{w}_t^{\eta_2}, \ldots$ into $\boldsymbol{w}_t$.

Surrogate Loss



Real loss

$$f_t(\boldsymbol{u}) \;\leq\; f_t(\boldsymbol{w}_t) + (\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \boldsymbol{g}_t$$

Surrogate loss

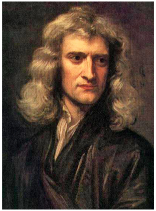
$$\ell_t^\eta(\boldsymbol{u}) \;:=\; \eta(\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \boldsymbol{g}_t + \left(\eta(\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \boldsymbol{g}_t\right)^2$$

Exp-concave! In particular:

$$e^{-\ell_t^\eta(\boldsymbol{u})} \;\leq\; 1 + \eta(\boldsymbol{w}_t - \boldsymbol{u})^\intercal \boldsymbol{g}_t.$$

Excellent bound $O(\ln T)$ for wrong loss.

MetaGrad Slave



η-Slave (variant of Online Newton Step) predicts

$$\boldsymbol{w}_{t+1}^{\eta} = \boldsymbol{w}_t^{\eta} - \eta \boldsymbol{\Sigma}_{t+1}^{\eta} \boldsymbol{g}_t$$

where the **covariance matrix** is given by

$$\boldsymbol{\Sigma}_{t+1}^{\eta} = \left(\frac{1}{4} \boldsymbol{I} + 2\eta^2 \sum_{s=1}^{t} \boldsymbol{g}_s \boldsymbol{g}_s^\intercal \right)^{-1}$$

η-Slave guarantees

$$\begin{aligned} \sum_{t=1}^{T} \left(\ell_t^{\eta}(\boldsymbol{w}_t^{\eta}) - \ell_t^{\eta}(\boldsymbol{u}) \right) &\leq \frac{1}{8} \|\boldsymbol{u}\|^2 + \frac{1}{2} \ln \det \left(\boldsymbol{I} + 8\eta^2 \sum_{t=1}^{T} \boldsymbol{g}_t \boldsymbol{g}_t^\intercal \right) \\ &\leq O(d \ln T) \quad \forall \boldsymbol{u} \end{aligned}$$

MetaGrad Master

Input: Grid points $\eta_i = 2^{-i}$ with weights $\pi_i = \frac{1}{i(i+1)}$.
Goal: aggregate $\boldsymbol{w}_t^{\eta_1}, \boldsymbol{w}_t^{\eta_2}, \ldots$

Idea: Potential

$$\Phi_t := \sum_i \pi_i e^{-\sum_{s=1}^t \ell_s^{\eta_i}(\boldsymbol{w}_s^{\eta_i})}.$$

Two steps:

- Find predictions $\boldsymbol{w}_t$ that ensure $1 \geq \Phi_1 \geq \Phi_2 \geq \ldots$
- Derive regret bound from $1 \geq \Phi_T$.

MetaGrad Master, Potential Decreases

Tilted exponentially weighted average

$$\boldsymbol{w}_{t+1} = \frac{\sum_i \pi_i e^{-\sum_{s=1}^t \ell_s^{\eta_i}(\boldsymbol{w}_s^{\eta_i})} \eta_i \boldsymbol{w}_{t+1}^{\eta_i}}{\sum_i \pi_i e^{-\sum_{s=1}^t \ell_s^{\eta_i}(\boldsymbol{w}_s^{\eta_i})} \eta_i}$$

ensures potential shrinks:

$$\begin{aligned}
\Phi_{t+1} - \Phi_t &= \sum_i \pi_i e^{-\sum_{s=1}^t \ell_s^{\eta_i}(\boldsymbol{w}_s^{\eta_i})} \left(e^{-\ell_{t+1}^{\eta_i}(\boldsymbol{w}_{t+1}^{\eta_i})} - 1 \right) \\
&\overset{\text{exp-con}}{\leq} \sum_i \pi_i e^{-\sum_{s=1}^t \ell_s^{\eta_i}(\boldsymbol{w}_s^{\eta_i})} \eta_i (\boldsymbol{w}_{t+1} - \boldsymbol{w}_{t+1}^{\eta_i})^\intercal \boldsymbol{g}_{t+1} \overset{\text{weights}}{=} 0
\end{aligned}$$

and hence $\Phi_t \leq 1$.

MetaGrad Master, Small Potential is Good

The Master achieves for all t:

$$1 \;\geq\; \Phi_t \;=\; \sum_i \pi_i e^{-\sum_{s=1}^t \ell_s^{\eta_i}(\boldsymbol{w}_s^{\eta_i})}.$$

It follows that

$$\sum_{t=1}^T \left(0 - \ell_t^{\eta_i}(\boldsymbol{w}_t^{\eta_i})\right) \;\leq\; -\ln \pi_i \qquad \forall i \text{ in grid}$$

(Master has zero surrogate loss)

MetaGrad Analysis

Now combine the Master and Slave guarantee.
For **each** grid point η and comparator $\boldsymbol{u}$

$$\sum_{t=1}^{T} \left(0 - \ell_t^{\eta}(\boldsymbol{w}_t^{\eta})\right) \;\leq\; -\ln \pi_i \;\leq\; \ln \ln T$$

$$\sum_{t=1}^{T} \left(\ell_t^{\eta}(\boldsymbol{w}_t^{\eta}) - \ell_t^{\eta}(\boldsymbol{u})\right) \;\leq\; O(d \ln T)$$

so

$$\sum_{t=1}^{T} \left(0 - \ell_t^{\eta}(\boldsymbol{u})\right) \;\leq\; O(d \ln T).$$

Unpacking $\ell_t^{\eta}(\boldsymbol{u}) = \eta(\boldsymbol{u} - \boldsymbol{w}_t)^{\intercal} \boldsymbol{g}_t + \left(\eta(\boldsymbol{u} - \boldsymbol{w}_t)^{\intercal} \boldsymbol{g}_t\right)^2$ yields

$$\eta R_T^{\boldsymbol{u}} \;\leq\; \eta^2 V_T^{\boldsymbol{u}} + O(d \ln T).$$

MetaGrad Analysis (ctd.)

Reorganise the bound to:

$$R_T^{\boldsymbol{u}} \;\leq\; \eta V_T^{\boldsymbol{u}} + \frac{O(d \ln T)}{\eta}$$

Now pick the best grid point

$$\hat{\eta} \;=\; \sqrt{\frac{O(d \ln T)}{V_T^{\boldsymbol{u}}}}$$

to find

$$R_T^{\boldsymbol{u}} \;\leq\; O\left(\sqrt{V_T^{\boldsymbol{u}} d \ln T}\right)$$

of course we need a grid point close to $\hat{\eta}$ and we need to deal with off-grid $\hat{\eta} \gg 1$ and $\hat{\eta} \ll \frac{1}{\sqrt{T}}$.

MetaGrad Outlook

- Run-time $O(d^2)$ per round
- Projections (avoid $O(d^3)$ per round!)
- We design and analyze two versions of Slave
  - Full covariance (quadratic)
  - Diagonal approximation (linear)
- Very welcome to discuss further

MetaGrad Outlook

- Run-time $O(d^2)$ per round
- Projections (avoid $O(d^3)$ per round!)
- We design and analyze two versions of Slave
  - Full covariance (quadratic)
  - Diagonal approximation (linear)
- Very welcome to discuss further

Learn more:

- Paper submitted to COLT 2016, preprint available
- Code is available

  `http://bitbucket.org/wmkoolen/metagrad`

- Experiments coming soon.

  `http://blog.wouterkoolen.info`

Summary

Low regret through stability, even without curvature.

- New MetaGrad algorithm.
- Hierarchical Master-Slave construction.
- Learns the learning rate.
- Refined (adaptive) regret bound.
- Stochastic condition for logarithmic regret (fast rates)

Thank you!