Combining Adversarial Guarantees and Fast Rates in Online Learning

http://bitbucket.org/wmkoolen/metagrad

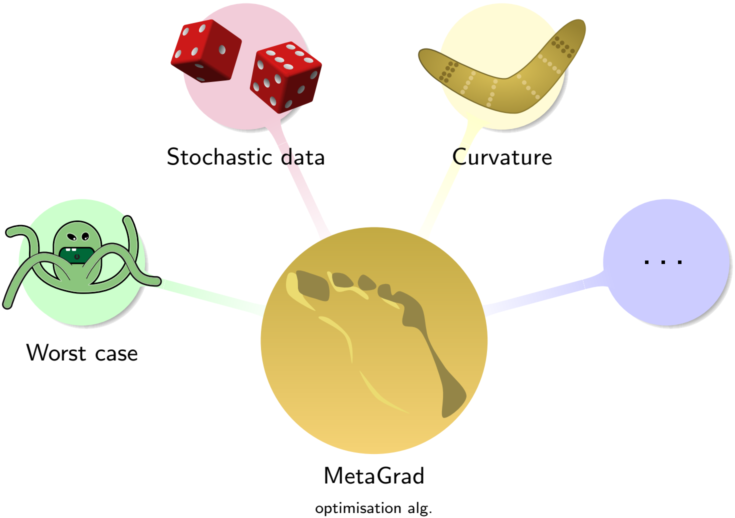
Wouter M. Koolen

CWI
Centrum Wiskunde & Informatica

Joint work with **Tim van Erven** and **Peter Grünwald**

Université Toulouse III - Paul Sabatier
Thursday 6$^{\text{th}}$ April, 2017

In a Nutshell



Menu

1. The online convex optimisation problem
2. State of the art
   - A taxonomy of losses
   - What's missing?
3. Main result: MetaGrad
   - Second order bound
   - Efficient implementation
4. Applications
   - Curvature
   - Stochastic case
   - Experiments

Optimisation Pervasive in Machine Learning

$$\min_{\boldsymbol{w}} \sum_{t=1}^{T} f_t(\boldsymbol{w})$$

Optimisation Pervasive in Machine Learning

$$\min_{w} \sum_{t=1}^{T} f_t(w)$$

Batch Training
(classification)

Optimisation Pervasive in Machine Learning

$$\min_{w} \sum_{t=1}^{T} f_t(w)$$

Batch Training
(classification)

Time Series
(investment)

Optimisation Pervasive in Machine Learning

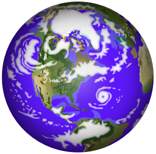
$$\min_{\boldsymbol{w}} \sum_{t=1}^{T} f_t(\boldsymbol{w})$$

Batch Training
(classification)

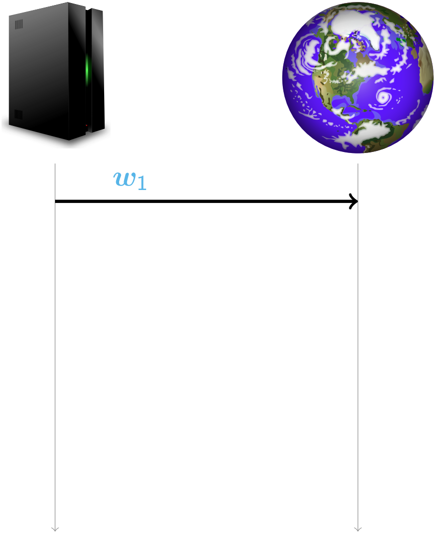
Time Series
(investment)

Big Data

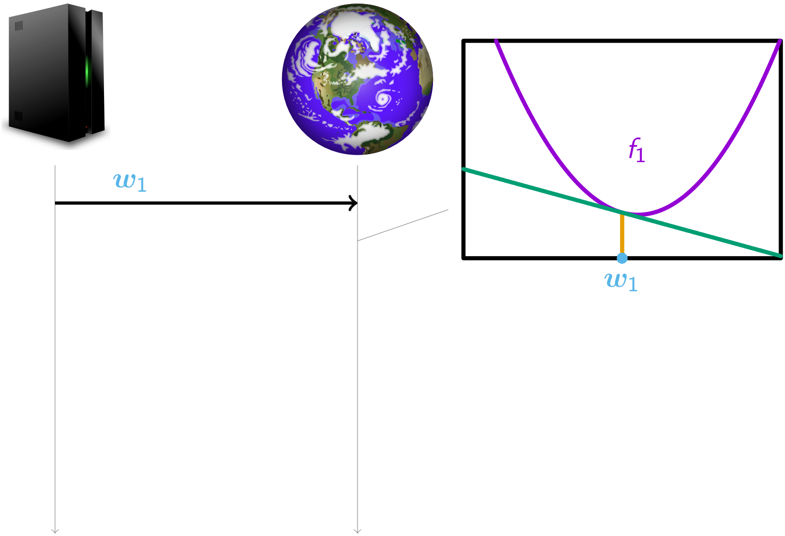
Online Convex Optimisation



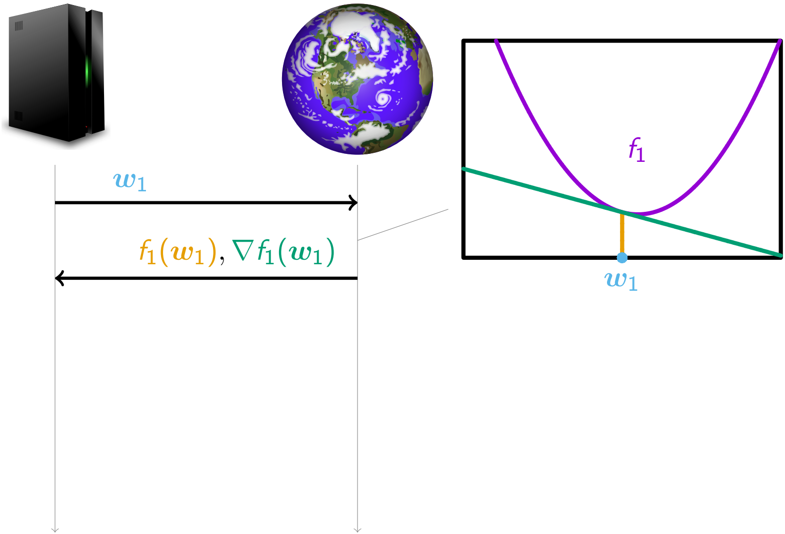
Online Convex Optimisation



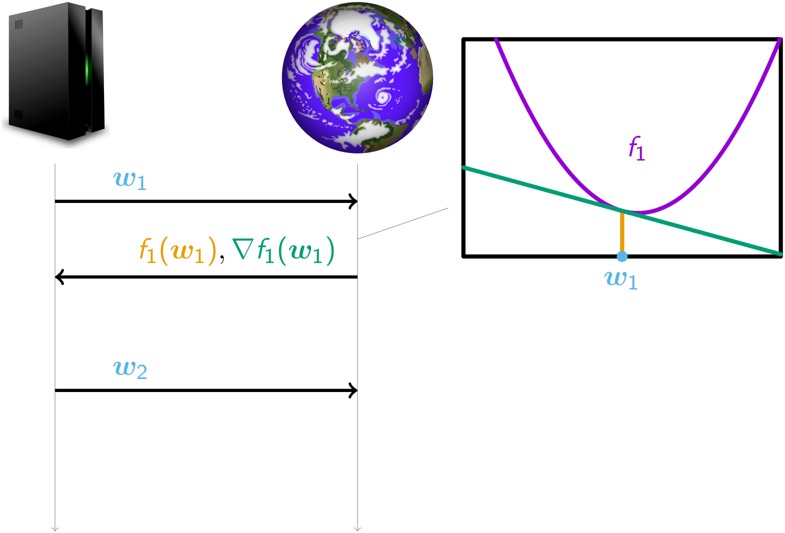
Online Convex Optimisation



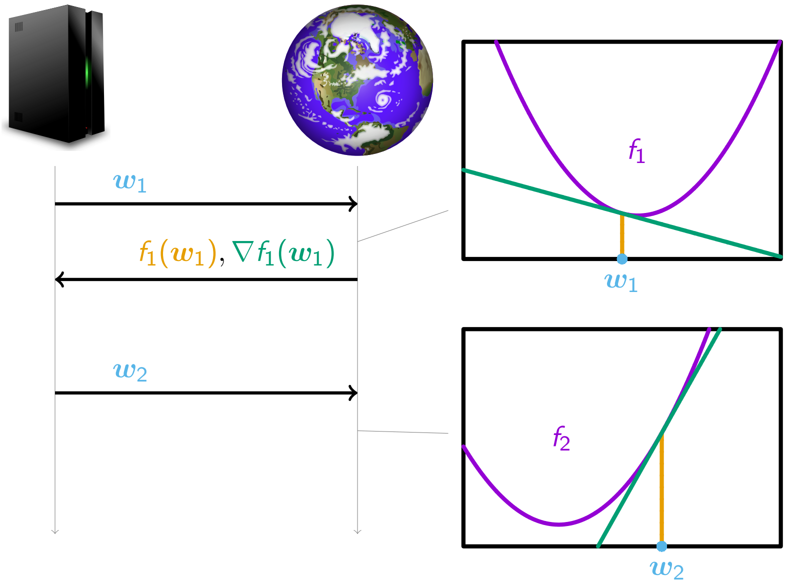
Online Convex Optimisation



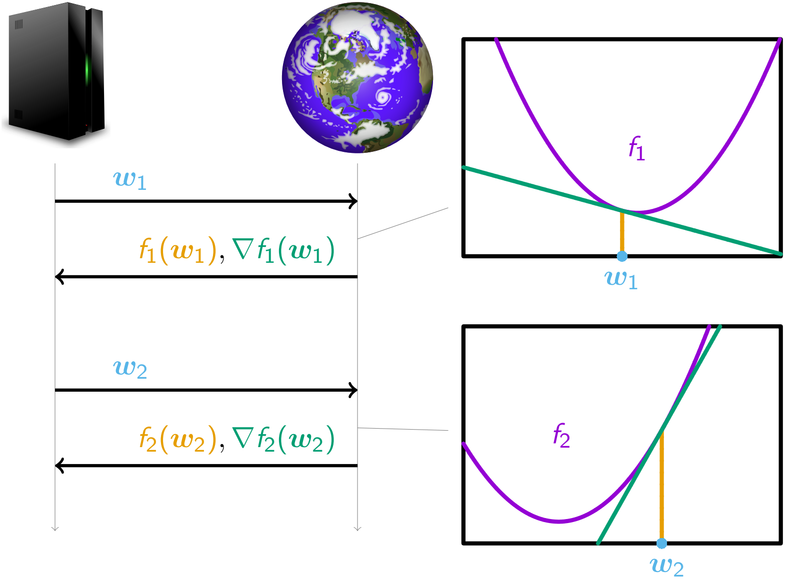
Online Convex Optimisation



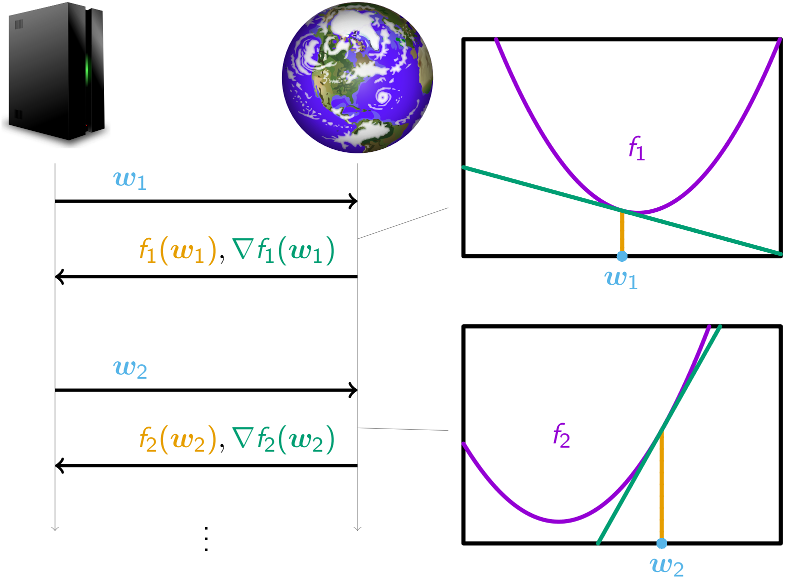
Online Convex Optimisation



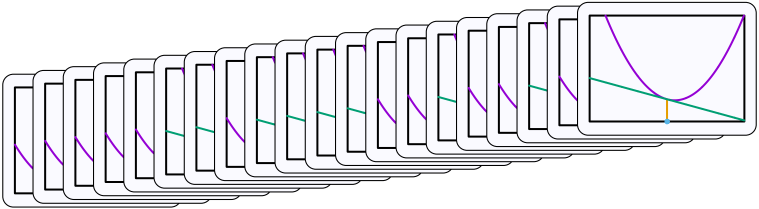
Online Convex Optimisation



Online Convex Optimisation



Objective



Definition (Regret)

$$R_T = \underbrace{\sum_{t=1}^{T} f_t(\boldsymbol{w}_t)}_{\text{Online loss}} - \underbrace{\min_{\boldsymbol{u}} \sum_{t=1}^{T} f_t(\boldsymbol{u})}_{\text{Optimal loss}}$$

Online Gradient Descent [Zinkevich, 2003]

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \nabla f_t(\boldsymbol{w}_t)$$

Online Gradient Descent [Zinkevich, 2003]

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \nabla f_t(\boldsymbol{w}_t)$$

Worst-case regret guarantee:

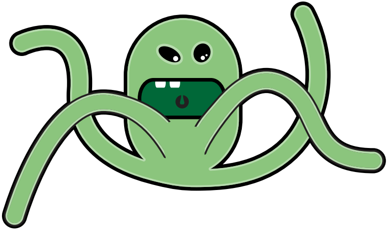
$$R_T = O\left(\sqrt{T}\right)$$

Online Gradient Descent [Zinkevich, 2003]

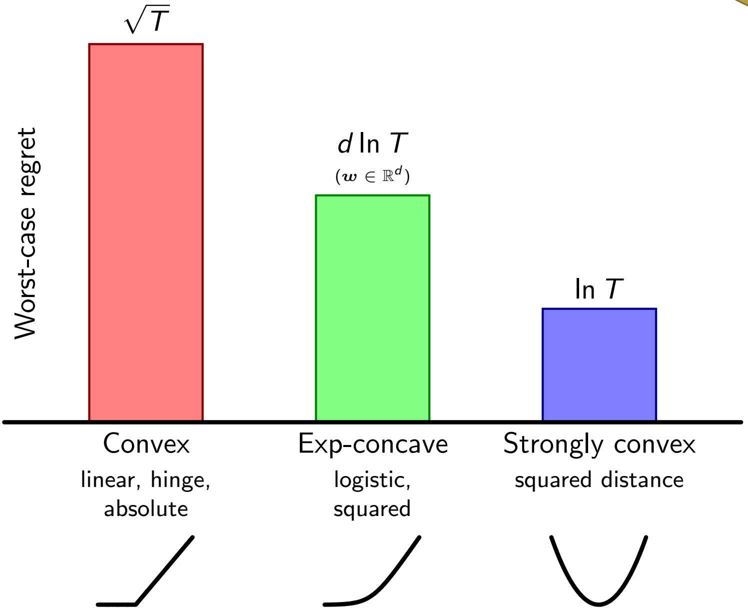
$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \nabla f_t(\boldsymbol{w}_t)$$

Worst-case regret guarantee:

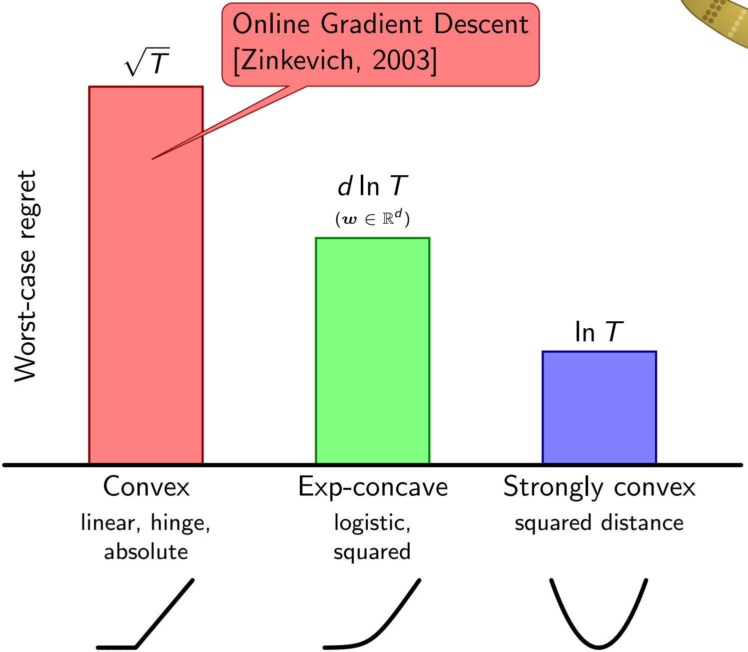
$$R_T = O\left(\sqrt{T}\right)$$



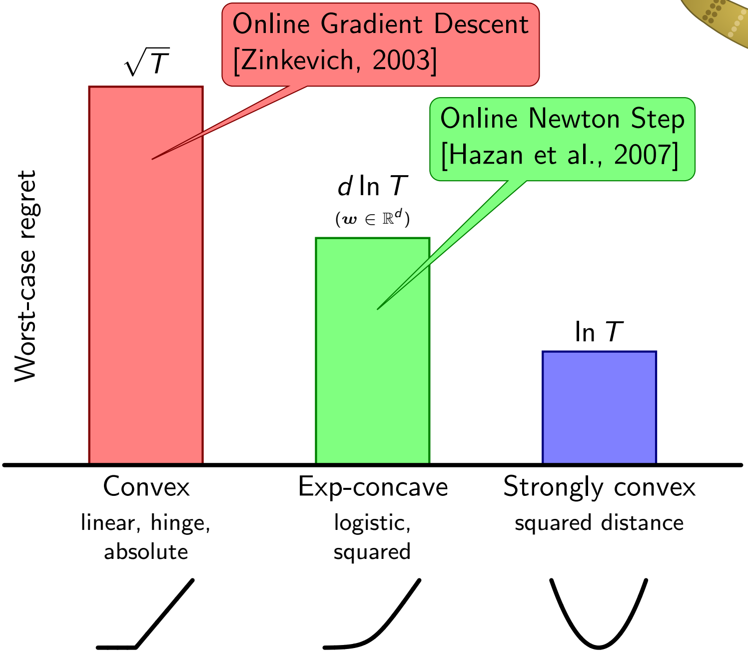
Loss Taxonomy $\sim$ Curvature



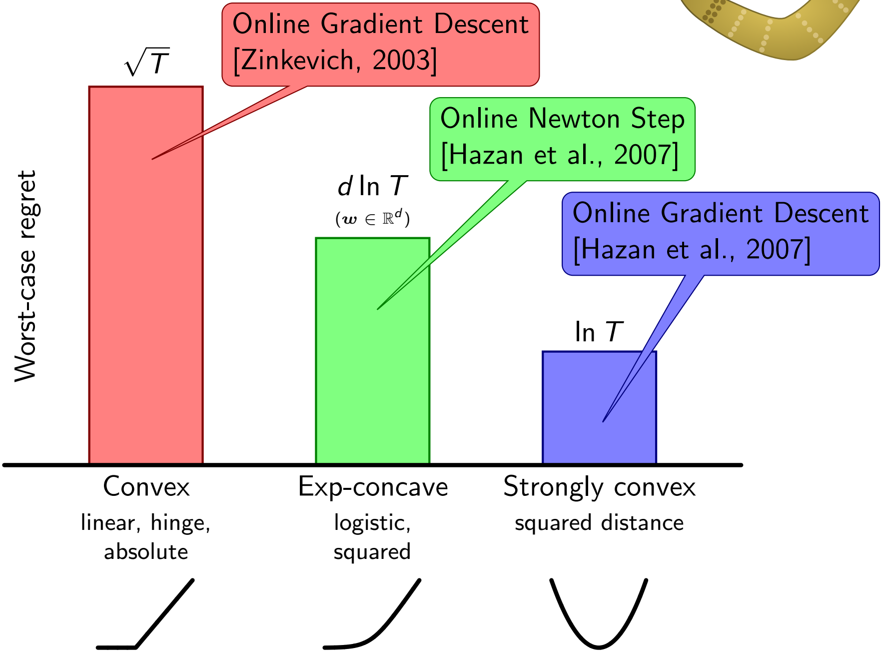
Loss Taxonomy ∼ Curvature



Loss Taxonomy ∼ Curvature



Loss Taxonomy ~ Curvature



Big Questions

Can we make **adaptive** methods for **online convex optimisation** that are

- **worst-case safe**
- exploit **curvature** automatically
- computationally **efficient**

Big Questions

Can we make **adaptive** methods for **online convex optimisation** that are

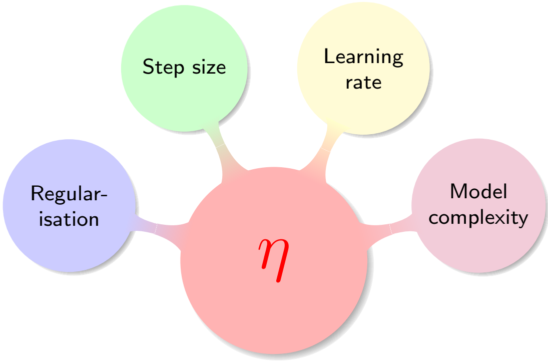
- **worst-case safe**
- exploit **curvature** automatically
- computationally **efficient**

And can we adapt to other important regimes?

- **Mixed** or **in-between** cases?
- **Stochastic** data? Bandits [Seldin and Slivkins, 2014]
- Absence of **curvature**? Experts [Koolen and Van Erven, 2015]

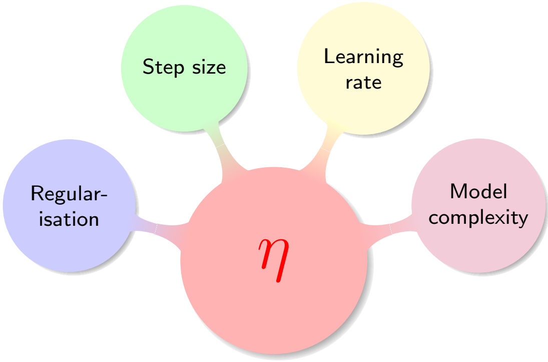
Main Idea

For every optimisation algorithm tuning is **crucial**.



Main Idea

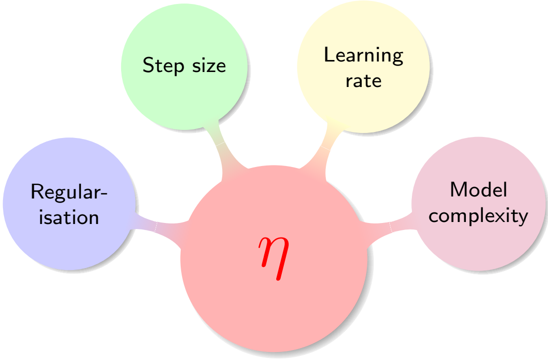
For every optimisation algorithm tuning is **crucial**.



So let's **learn** optimal tuning from **data**.

Main Idea

For every optimisation algorithm tuning is **crucial**.

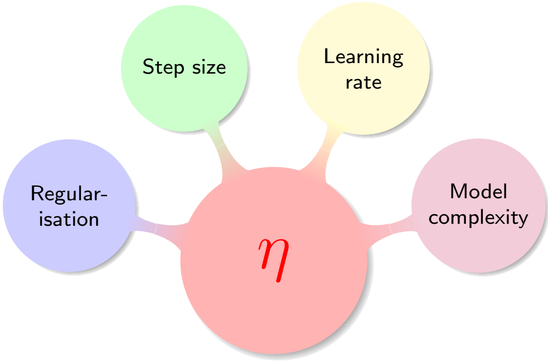


So let's **learn** optimal tuning from **data**.

Key obstacle: avoid learning η at **slow rate** itself.

Main Idea

For every optimisation algorithm tuning is **crucial**.

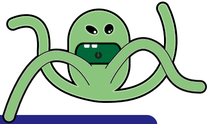


So let's **learn** optimal tuning from **data**.

Key obstacle: avoid learning η at **slow rate** itself.

Breakthrough: Multiple Eta Gradient algorithm (MetaGrad)

Second-order Regret Bound



Theorem

The regret of MetaGrad is bounded by

$$R_T = O\left(\min\left\{\sqrt{T}, \sqrt{V_T\, d \ln T}\right\}\right),$$

where

$$V_T = \sum_{t=1}^{T} \left((\boldsymbol{w}_t - \boldsymbol{u}^*)^\intercal \nabla f_t(\boldsymbol{w}_t)\right)^2$$

measures **variance** *compared to the* **offline optimum** $\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}} \sum_{t=1}^{T} f_t(\boldsymbol{u})$

Note: Optimal tuning depends on unknown optimum $\boldsymbol{u}^*$.

Proof Ideas

Analysis based on second-order surrogate loss. For each η:

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

Proof Ideas

Analysis based on second-order surrogate loss. For each η:

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

Since surrogate is exp-concave for each fixed η, we can use **online quasi-Newton method** like Online Newton Step [Hazan et al., 2007] to get predictions $\boldsymbol{w}_t^{\eta}$ that achieve logarithmic regret:

$$\sum_{t=1}^{T} \ell_t^{\eta}(\boldsymbol{w}_t^{\eta}) - \sum_{t=1}^{T} \ell_t^{\eta}(\boldsymbol{u}) \ \leq \ O(d \ln T) \qquad \forall \boldsymbol{u} \in \mathcal{U}$$

Proof Ideas

Analysis based on second-order surrogate loss. For each η:

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

Since surrogate is exp-concave for each fixed η, we can use **online quasi-Newton method** like Online Newton Step [Hazan et al., 2007] to get predictions $\boldsymbol{w}_t^\eta$ that achieve logarithmic regret:

$$\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta) - \sum_{t=1}^T \ell_t^\eta(\boldsymbol{u}) \;\leq\; O(d \ln T) \qquad \forall \boldsymbol{u} \in \mathcal{U}$$

To learn the best η we combine the predictions $\boldsymbol{w}_t^\eta$ for multiple η into a single master prediction $\boldsymbol{w}_t$ using an **experts algorithm for combining multiple learning rates** similar to Squint [Koolen and Van Erven, 2015], to get:

$$\underbrace{\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t)}_{=0} - \sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta) \;\leq\; O(\ln \ln T) \qquad \forall \eta$$

Proof Ideas

Analysis based on second-order surrogate loss. For each η:

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

Since surrogate is exp-concave for each fixed η, we can use **online quasi-Newton method** like Online Newton Step [Hazan et al., 2007] to get predictions $\boldsymbol{w}_t^\eta$ that achieve logarithmic regret:

$$\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta) - \sum_{t=1}^T \ell_t^\eta(\boldsymbol{u}) \;\le\; O(d \ln T) \qquad \forall \boldsymbol{u} \in \mathcal{U}$$

To learn the best η we combine the predictions $\boldsymbol{w}_t^\eta$ for multiple η into a single master prediction $\boldsymbol{w}_t$ using an **experts algorithm for combining multiple learning rates** similar to Squint [Koolen and Van Erven, 2015], to get:

$$\underbrace{\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t)}_{=0} - \sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta) \;\le\; O(\ln \ln T) \qquad \forall \eta$$

Difficulty: Master has to perform well under multiple loss functions simultaneously. No standard experts algorithm works!

Proof Ideas

Analysis based on second-order surrogate loss. For each η:

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

Since surrogate is exp-concave for each fixed η, we can use **online quasi-Newton method** like Online Newton Step [Hazan et al., 2007] to get predictions $\boldsymbol{w}_t^\eta$ that achieve logarithmic regret:

$$\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta) - \sum_{t=1}^T \ell_t^\eta(\boldsymbol{u}) \; \leq \; O(d \ln T) \qquad \forall \boldsymbol{u} \in \mathcal{U}$$

To learn the best η we combine the predictions $\boldsymbol{w}_t^\eta$ for multiple η into a single master prediction $\boldsymbol{w}_t$ using an **experts algorithm for combining multiple learning rates** similar to Squint [Koolen and Van Erven, 2015], to get:

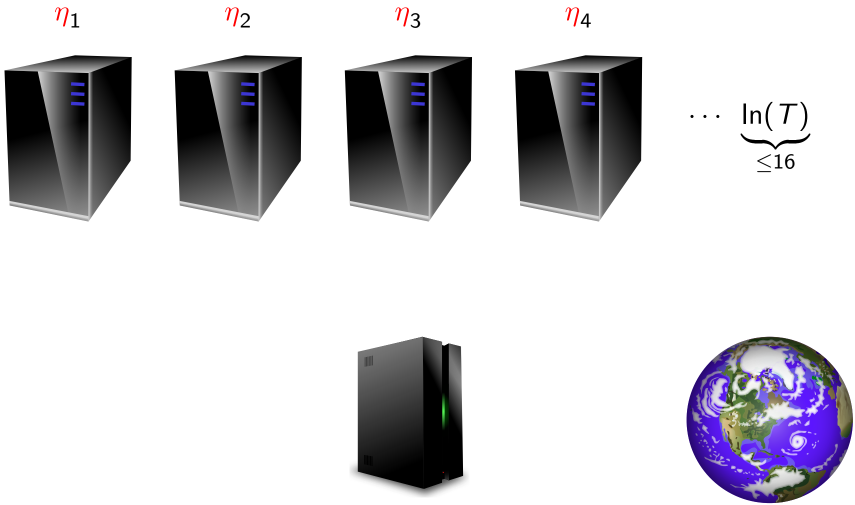
$$\underbrace{\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t)}_{=0} - \sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta) \; \leq \; O(\ln \ln T) \qquad \forall \eta$$

Difficulty: Master has to perform well under multiple loss functions simultaneously. No standard experts algorithm works!

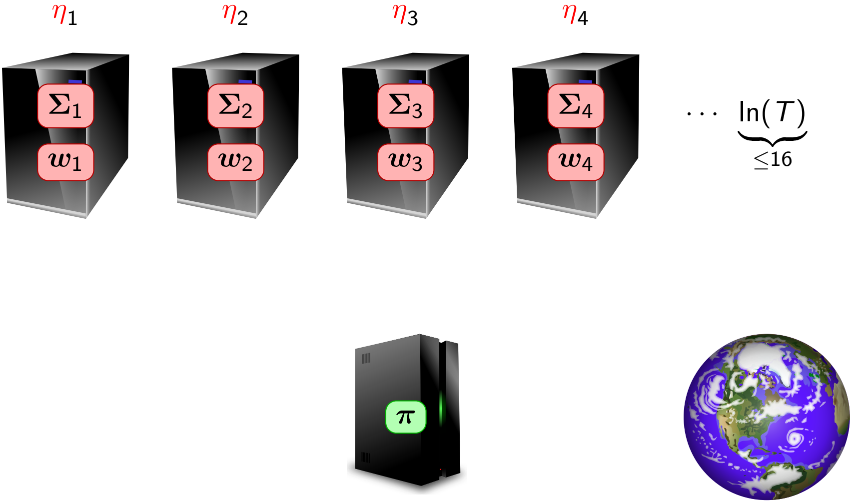
Together: $-\sum_{t=1}^T \ell_t^\eta(\boldsymbol{u}) \leq O(d \ln T)$ for each η and $\boldsymbol{u}$, resulting in

$$R_T \; \leq \; \sum_{t=1}^T (\boldsymbol{w}_t - \boldsymbol{u})^\intercal \boldsymbol{g}_t \; \leq \; \frac{O(d \ln T)}{\eta} + \eta V_T^{\boldsymbol{u}} \; \Rightarrow \; O\left(\sqrt{V_T^{\boldsymbol{u}} d \ln T}\right).$$

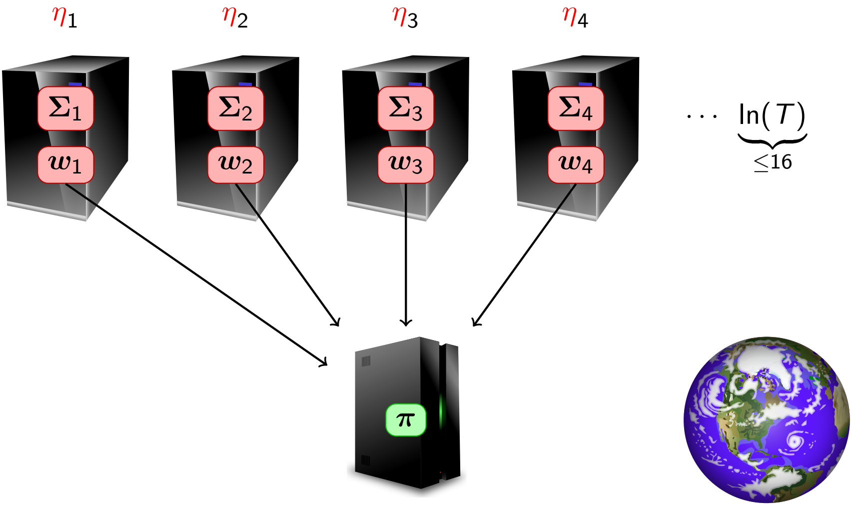
MetaGrad Algorithm



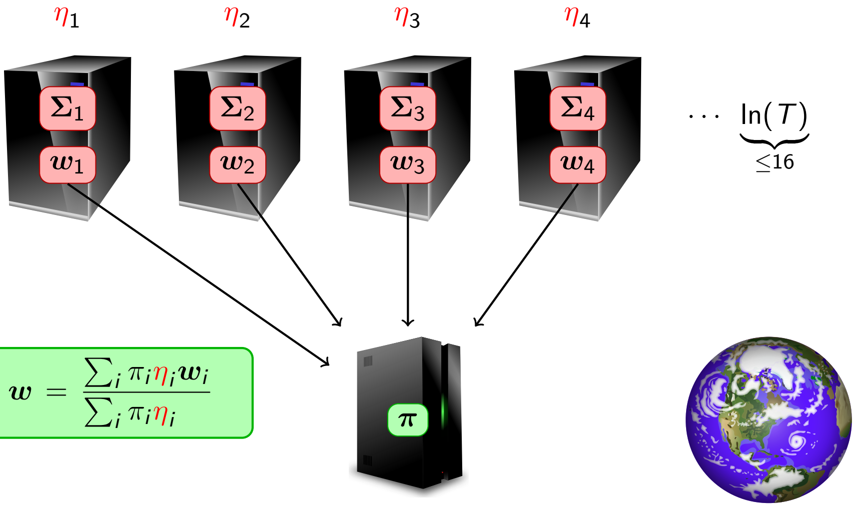
MetaGrad Algorithm



MetaGrad Algorithm

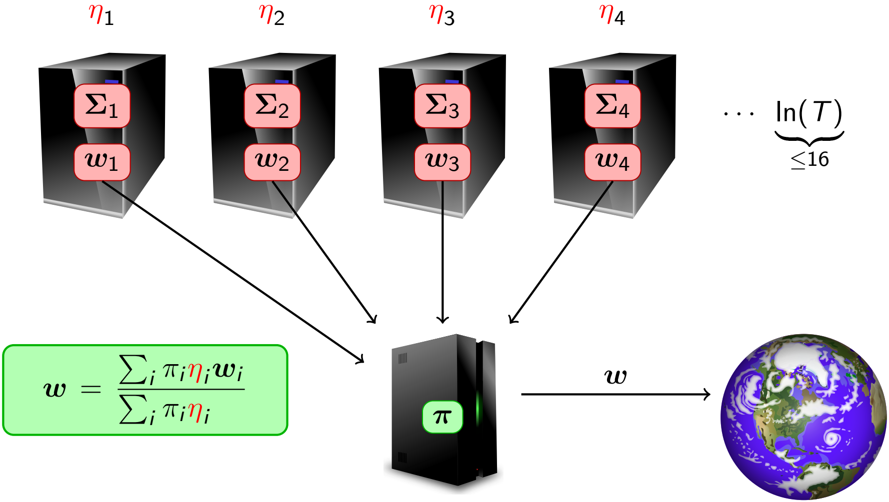


MetaGrad Algorithm

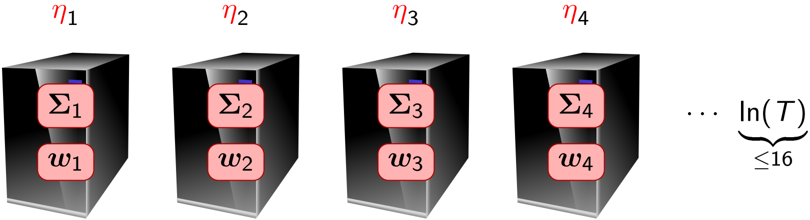


$$\boldsymbol{w} = \frac{\sum_i \pi_i \eta_i \boldsymbol{w}_i}{\sum_i \pi_i \eta_i}$$

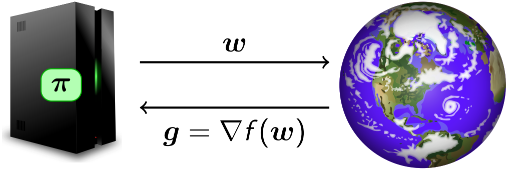
MetaGrad Algorithm



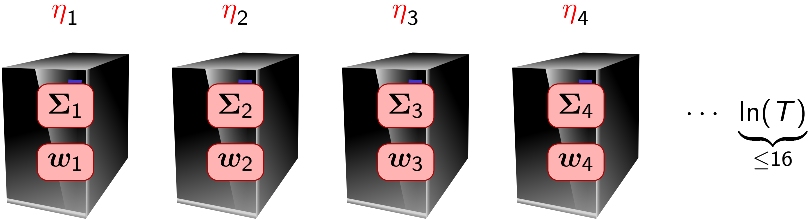
MetaGrad Algorithm



$$\boldsymbol{w} = \frac{\sum_i \pi_i \eta_i \boldsymbol{w}_i}{\sum_i \pi_i \eta_i}$$



MetaGrad Algorithm

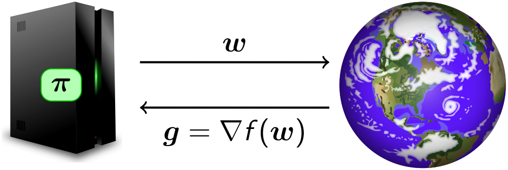


$$\boldsymbol{w} = \frac{\sum_i \pi_i \eta_i \boldsymbol{w}_i}{\sum_i \pi_i \eta_i}$$

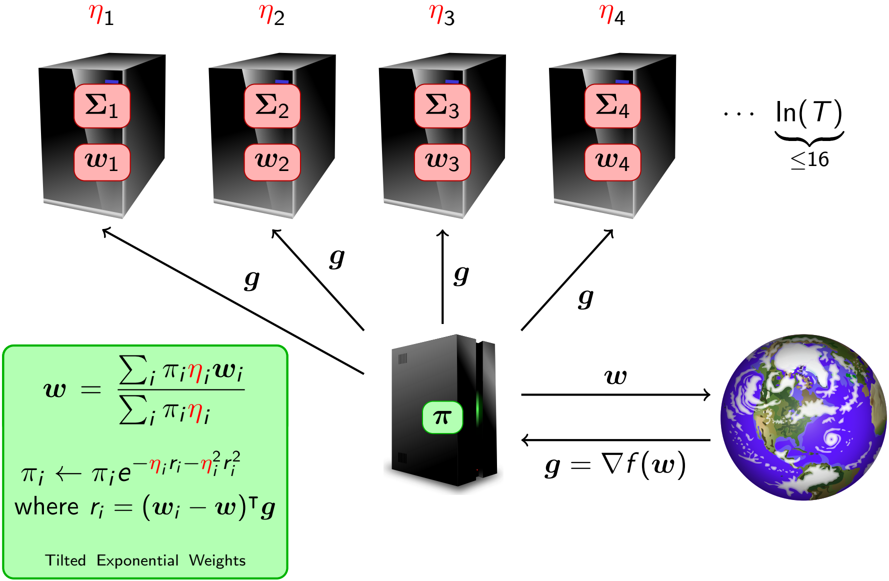
$$\pi_i \leftarrow \pi_i e^{-\eta_i r_i - \eta_i^2 r_i^2}$$

where $r_i = (\boldsymbol{w}_i - \boldsymbol{w})^\intercal \boldsymbol{g}$

Tilted Exponential Weights



MetaGrad Algorithm

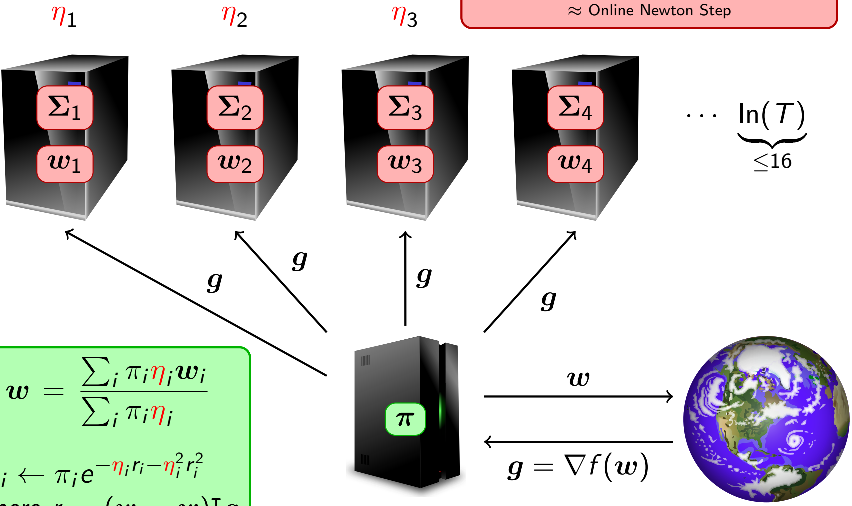


MetaGrad Algorithm

$$\boldsymbol{\Sigma}_i \leftarrow (\boldsymbol{\Sigma}_i^{-1} + 2\eta_i^2 \boldsymbol{g}\boldsymbol{g}^\intercal)^{-1}$$

$$\boldsymbol{w}_i \leftarrow \boldsymbol{w}_i - \eta_i \boldsymbol{\Sigma}_i \boldsymbol{g}\,(1 + 2\eta_i r_i)$$

≈ Online Newton Step



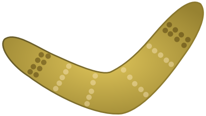
$$\boldsymbol{w} = \frac{\sum_i \pi_i \eta_i \boldsymbol{w}_i}{\sum_i \pi_i \eta_i}$$

$$\pi_i \leftarrow \pi_i e^{-\eta_i r_i - \eta_i^2 r_i^2}$$

where $r_i = (\boldsymbol{w}_i - \boldsymbol{w})^\intercal \boldsymbol{g}$

Tilted Exponential Weights

MetaGrad Adapts to Curvature



MetaGrad regret bound:

$$R_T \;=\; O\left(\sqrt{V_T\, d \ln T}\right)$$

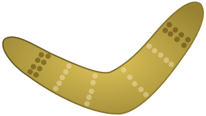
Corollary

For α-**exp-concave** *or* α-**strongly convex** *losses, MetaGrad ensures*

$$R_T \;=\; O\left(d \ln T\right)$$

without knowing α.

MetaGrad Adapts to Curvature



MetaGrad regret bound:

$$R_T = O\left(\sqrt{V_T \, d \ln T}\right)$$

Corollary

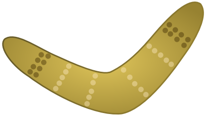
*For α-**exp-concave** or α-**strongly convex** losses, MetaGrad ensures*

$$R_T = O(d \ln T)$$

without knowing α.

Same result for fixed $f_t = f$ (classical optimisation) even without curvature via **derivative condition**.

MetaGrad Adapts to Curvature



MetaGrad regret bound:

$$R_T \;=\; O\left(\sqrt{V_T\, d \ln T}\right)$$

Corollary

For α-**exp-concave** *or* α-**strongly convex** *losses, MetaGrad ensures*

$$R_T \;=\; O\left(d \ln T\right)$$

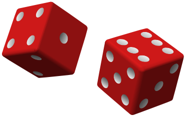
without knowing α*.*

Same result for fixed $f_t = f$ (classical optimisation) even without curvature via **derivative condition**.

Reason

Curvature implies $\Omega(V_T)$ cumulative slack between loss and its tangent lower bound.

MetaGrad Adapts to Stochastic Margin



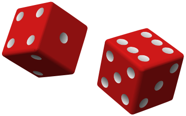
Consider i.i.d. losses $f_t \sim \mathbb{P}$ with **stochastic optimum**

$$\boldsymbol{u}^* \;=\; \arg\min_{\boldsymbol{u}} \mathbb{E}\, f(\boldsymbol{u})$$

Goal is small **pseudo-regret** compared to $\boldsymbol{u}^*$:

$$R_T^* \;=\; \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \sum_{t=1}^{T} f_t(\boldsymbol{u}^*)$$

MetaGrad Adapts to Stochastic Margin



Consider i.i.d. losses $f_t \sim \mathbb{P}$ with **stochastic optimum**

$$\boldsymbol{u}^* \;=\; \arg\min_{\boldsymbol{u}} \mathbb{E}\, f(\boldsymbol{u})$$

Goal is small **pseudo-regret** compared to $\boldsymbol{u}^*$:

$$R_T^* \;=\; \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \sum_{t=1}^{T} f_t(\boldsymbol{u}^*)$$

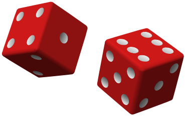
Corollary (with Peter Grünwald)

For any β-**Bernstein** $\mathbb{P}$, *MetaGrad keeps the expected regret below*

$$\mathbb{E}\, R_T^* \;\leq\; O\left((d \ln T)^{\frac{1}{2-\beta}}\; T^{\frac{1-\beta}{2-\beta}}\right).$$

Fast rates without curvature: e.g. absolute loss, hinge loss, ...

MetaGrad Adapts to Stochastic Margin



Consider i.i.d. losses $f_t \sim \mathbb{P}$ with **stochastic optimum**

$$\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}} \mathbb{E}\, f(\boldsymbol{u})$$

Goal is small **pseudo-regret** compared to $\boldsymbol{u}^*$:

$$R_T^* = \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \sum_{t=1}^{T} f_t(\boldsymbol{u}^*)$$

Corollary (with Peter Grünwald)

For any β-**Bernstein** $\mathbb{P}$, *MetaGrad keeps the expected regret below*

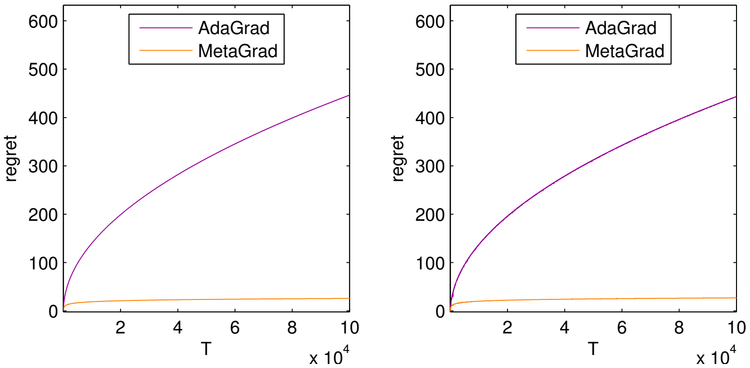
$$\mathbb{E}\, R_T^* \leq O\left((d \ln T)^{\frac{1}{2-\beta}}\, T^{\frac{1-\beta}{2-\beta}}\right).$$

Fast rates without curvature: e.g. absolute loss, hinge loss, ...

Reason

Bernstein bounds $\mathbb{E}[V_T^*]$ above by $\mathbb{E}[R_T^*]$. "Solve" regret bound.

Experiments



(a) Offline: $f_t(u) = |u - 1/4|$

(b) Stochastic Online: $f_t(u) = |u - x_t|$ where $x_t = \pm\frac{1}{2}$ i.i.d. with probabilities 0.4 and 0.6.

Figure: Examples of fast rates on functions without curvature. MetaGrad incurs logarithmic regret $O(\log T)$, while AdaGrad incurs $O(\sqrt{T})$ regret, matching its bound.

Conclusion

First contact with a new generation of **adaptive algorithms**.

Conclusion

First contact with a new generation of **adaptive algorithms**.

MetaGrad adapts to a wide range of environments: