MetaGrad: Adapting to the Distribution of the Data in Online Sequential Prediction

Tim van Erven

Universiteit Leiden

Joint work with: Wouter Koolen, Peter Grünwald

Rennes, March 17, 2017

Offline vs Online Learning

I. (Offline) Statistical Learning

- Limits of minimax analysis pretty well understood (margin/Bernstein condition)
- Adaptive algorithms known to exploit 'easy' data distributions

II. Online Sequential Prediction

- Most results still about minimax analysis
- We develop theory of adaptive analysis:
  - New online version of the Bernstein condition
  - New adaptive algorithm: MetaGrad
  - Adaptively achieving (optimal) fast rates

Part I: Statistical Learning Background

Statistical Learning

$$\begin{pmatrix} Y_1 \\ \boldsymbol{X}_1 \end{pmatrix}, \ldots, \begin{pmatrix} Y_N \\ \boldsymbol{X}_N \end{pmatrix} \qquad \text{independently distributed } \sim P$$

$$\downarrow$$

$$\hat{f} \in \mathcal{F}$$

$$\downarrow$$

small risk $\qquad R(\hat{f}) = \mathop{\mathbb{E}}_{(\boldsymbol{X},Y)\sim P}[\text{loss}(\boldsymbol{X}, Y, \hat{f})]$

compared to minimizer f^* of risk in model $\mathcal{F}$

Minimax Rate:

Rate for most difficult possible P

$$\min_{\hat{f}} \max_{P} \; \mathbb{E}[R(\hat{f})] - R(f^*)$$

Classification

Given $\boldsymbol{X} \in \mathbb{R}^d$, predict binary label $Y \in \{0, 1\}$

$$\text{loss}(\boldsymbol{X}, Y, f) = \begin{cases} 0 & \text{if } f(\boldsymbol{X}) = Y, \\ 1 & \text{if } f(\boldsymbol{X}) \neq Y \end{cases}$$

$$R(f) = P(f(\boldsymbol{X}) \neq Y)$$

Minimax Rate:

For worst-case P, learning is slow:

$$\mathbb{E}[R(\hat{f})] - R(f^*) \precsim \sqrt{\frac{\text{complexity}(\mathcal{F})}{N}}$$

But Faster Rates Are Common

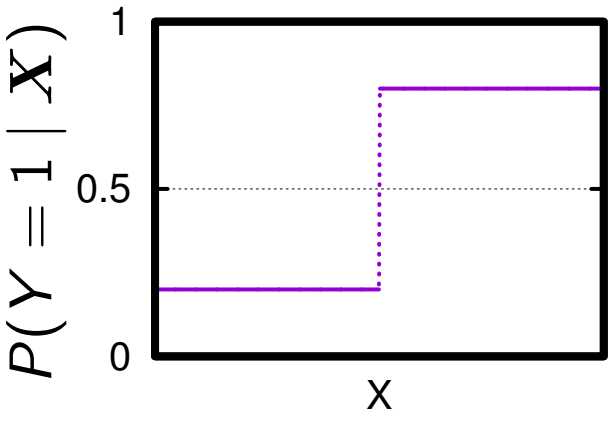
- Worst-case distribution: $P(Y = 1 \mid \boldsymbol{X})$ very close to $\frac{1}{2}$
- But then learning is (almost) useless!

The Margin Condition: [Tsybakov, 2004]

- Common case: $P(Y = 1 \mid \boldsymbol{X})$ not too close to $\frac{1}{2}$
- Assume $f^*(\boldsymbol{X}) = f_{\mathrm{B}}(\boldsymbol{X}) = \arg\max_y P(Y = y \mid \boldsymbol{X})$
- Then learning is much faster, up to

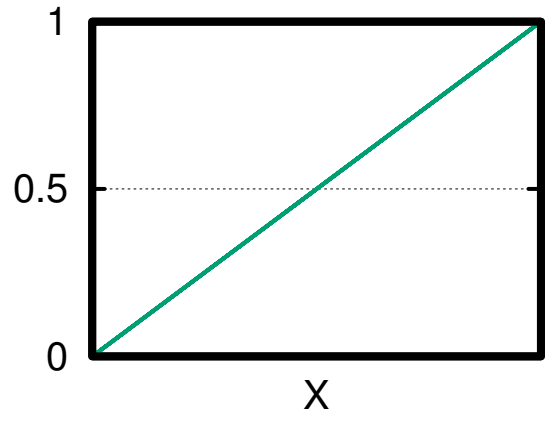
$$\mathbb{E}[R(\hat{f})] - R(f^*) \preccurlyeq \frac{\text{complexity}(\mathcal{F})}{N}$$

The Margin Condition



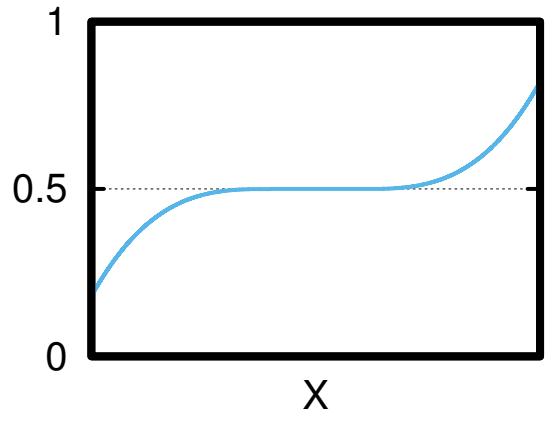
easy

$\alpha = \infty$



moderate

$\alpha = 1$



hard

$\alpha = 0$

$$P_{\boldsymbol{X}}\left(|P(Y \mid \boldsymbol{X}) - \tfrac{1}{2}| \leq t\right) \leq c\, t^{\alpha}$$

Fast Rates Because Margin Reduces Variance

Important source of excess risk $R(\hat{f}) - R(f^*)$ is **variance in excess loss**:

$$V(\hat{f}, f^*) = \mathbb{E}\left(\text{loss}(\boldsymbol{X}, Y, \hat{f}) - \text{loss}(\boldsymbol{X}, Y, f^*)\right)^2$$

Lemma (Tsybakov)

*If $f^* = f_B$. Then α-margin is equivalent to the (B, β)-**Bernstein condition***

$$V(\hat{f}, f^*) \leq B\left(R(\hat{f}) - R(f^*)\right)^{\beta}$$

for $\beta = \frac{\alpha}{1+\alpha} \in [0, 1]$ and some constant $B \geq 0$.

Fast Rates Because Margin Reduces Variance

Important source of excess risk $R(\hat{f}) - R(f^*)$ is **variance in excess loss**:

$$V(\hat{f}, f^*) = \mathbb{E}\left(\text{loss}(\boldsymbol{X}, Y, \hat{f}) - \text{loss}(\boldsymbol{X}, Y, f^*)\right)^2$$

Lemma (Tsybakov)

*If $f^* = f_B$. Then α-margin is equivalent to the (B, β)-*__Bernstein condition__

$$V(\hat{f}, f^*) \leq B\left(R(\hat{f}) - R(f^*)\right)^{\beta}$$

for $\beta = \frac{\alpha}{1+\alpha} \in [0, 1]$ and some constant $B \geq 0$.

Fast Rates Because Margin Reduces Variance

Important source of excess risk $R(\hat{f}) - R(f^*)$ is **variance in excess loss**:

$$V(\hat{f}, f^*) = \mathbb{E}\left(\text{loss}(\boldsymbol{X}, Y, \hat{f}) - \text{loss}(\boldsymbol{X}, Y, f^*)\right)^2$$

Lemma (Tsybakov)

*If $f^* = f_B$. Then α-margin is equivalent to the (B, β)-*__Bernstein condition__

$$V(\hat{f}, f^*) \leq B\left(R(\hat{f}) - R(f^*)\right)^{\beta}$$

for $\beta = \frac{\alpha}{1+\alpha} \in [0, 1]$ and some constant $B \geq 0$.

Adaptive Statistical Learning

- For simplicity: **prior π** on **countable model** $\mathcal{F} = \{f_1, f_2, \ldots\}$.
- **Penalized ERM** $\hat{f}$ minimizes

$$\sum_{i=1}^{N} \text{loss}(\boldsymbol{X}_i, Y_i, f) + \lambda \log \frac{1}{\pi(f)}$$

Crucial question: **how to tune λ?**

Adaptive Statistical Learning

- For simplicity: **prior** π on **countable model** $\mathcal{F} = \{f_1, f_2, \ldots\}$.
- **Penalized ERM** $\hat{f}$ minimizes

$$\sum_{i=1}^{N} \text{loss}(\boldsymbol{X}_i, Y_i, f) + \lambda \log \frac{1}{\pi(f)}$$

Proposition (Bernstein Condition Rate)

(Assuming bounded loss.) Knowing β, $\lambda = \left(\frac{N}{\log \frac{1}{\pi(f^*)}}\right)^{\frac{1-\beta}{2-\beta}}$ *achieves*

$$R(\hat{f}) - R(f^*) \preccurlyeq \left(\frac{\log \frac{1}{\delta\pi(f^*)}}{N}\right)^{\frac{1}{2-\beta}} \qquad \text{w.p.} \geq 1 - \delta.$$

Adaptive Statistical Learning

- For simplicity: **prior** π on **countable model** $\mathcal{F} = \{f_1, f_2, \ldots\}$.
- **Penalized ERM** $\hat{f}$ minimizes

$$\sum_{i=1}^{N} \text{loss}(\boldsymbol{X}_i, Y_i, f) + \lambda \log \frac{1}{\pi(f)}$$

Proposition (Bernstein Condition Rate)

(Assuming bounded loss.) Knowing β, $\lambda = \left(\frac{N}{\log \frac{1}{\pi(f^*)}}\right)^{\frac{1-\beta}{2-\beta}}$ *achieves*

$$R(\hat{f}) - R(f^*) \preccurlyeq \left(\frac{\log \frac{1}{\delta\pi(f^*)}}{N}\right)^{\frac{1}{2-\beta}} \qquad \text{w.p.} \geq 1 - \delta.$$

- **Simple adaptive** method: estimate λ on hold out set
- Or **very sophisticated adaptive approaches**:
  - Slope heuristic (Birgé, Massart)
  - Lepski's method
  - Safe Bayes (Grünwald)

Part II: Online Sequential Prediction

- Same problem of tuning $\lambda \leftrightarrow 1/\eta$
- Hold-out estimation not possible
- Very limited adaptive methods and theory

Part II: Online Sequential Prediction

- Same problem of tuning $\lambda \leftrightarrow 1/\eta$
- Hold-out estimation not possible
- Very limited adaptive methods and theory
- Our contribution: develop theory and solve tuning problem

Online Sequential Prediction

Online Data

- Data $\begin{pmatrix} Y_1 \\ \boldsymbol{X}_1 \end{pmatrix}, \ldots, \begin{pmatrix} Y_T \\ \boldsymbol{X}_T \end{pmatrix}$ arrive one by one, sequentially
- No assumption that data follow any distribution

Online Estimation

- Focus on parametric models $\mathcal{F} = \{f_{\boldsymbol{w}} \mid \boldsymbol{w} \in \mathbb{R}^d\}$, but often high dimensional
- Estimate $\boldsymbol{w}_t$ from $t-1$ examples, predict Y_t by $f_{\boldsymbol{w}_t}(\boldsymbol{X}_t)$

Notation

- Abbreviate $\ell_t(\boldsymbol{w}) = \text{loss}(\boldsymbol{X}_t, Y_t, f_{\boldsymbol{w}})$

Online Convex Optimization

Parameters $\boldsymbol{w}$ take values in a convex domain $\mathcal{U}$

1: **for** $t = 1, 2, \ldots, T$ **do**

2: Learner plays $\boldsymbol{w}_t \in \mathcal{U}$

3: Environment reveals convex loss function $\ell_t : \mathcal{U} \to \mathbb{R}$

4: Learner incurs loss $\ell_t(\boldsymbol{w}_t)$, observes gradient $\boldsymbol{g}_t = \nabla \ell_t(\boldsymbol{w}_t)$

5: **end for**

Online Convex Optimization

Parameters $\boldsymbol{w}$ take values in a convex domain $\mathcal{U}$

1: **for** $t = 1, 2, \ldots, T$ **do**

2: Learner plays $\boldsymbol{w}_t \in \mathcal{U}$

3: Environment reveals convex loss function $\ell_t : \mathcal{U} \to \mathbb{R}$

4: Learner incurs loss $\ell_t(\boldsymbol{w}_t)$, observes gradient $\boldsymbol{g}_t = \nabla \ell_t(\boldsymbol{w}_t)$

5: **end for**

Example: Classification with Convex Surrogate Losses

Given bounded $\boldsymbol{X}_t \in [-1, +1]^d$, predict label $Y_t \in \{-1, 1\}$

$$\ell_t(\boldsymbol{w}) = \max\{0, 1 - Y_t \langle \boldsymbol{w}, \boldsymbol{X}_t \rangle\} \qquad \text{(hinge loss)}$$

$$\ell_t(\boldsymbol{w}) = \ln\left(1 + e^{-Y_t \langle \boldsymbol{w}, \boldsymbol{X}_t \rangle}\right) \qquad \text{(logistic loss)}$$

$$\ell_t(\boldsymbol{w}) = (Y_t - \langle \boldsymbol{w}, \boldsymbol{X}_t \rangle)^2 \qquad \text{(squared loss)}$$

for $\boldsymbol{w}$ in Euclidean ball with diameter D.

Online Convex Optimization

Parameters $\boldsymbol{w}$ take values in a convex domain $\mathcal{U}$

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Learner plays $\boldsymbol{w}_t \in \mathcal{U}$
3: Environment reveals convex loss function $\ell_t : \mathcal{U} \to \mathbb{R}$
4: Learner incurs loss $\ell_t(\boldsymbol{w}_t)$, observes gradient $\boldsymbol{g}_t = \nabla \ell_t(\boldsymbol{w}_t)$
5: **end for**

Minimize **regret** w.r.t. oracle parameters $\boldsymbol{u} \in \mathcal{U}$:

$$\text{Regret}_T^{\boldsymbol{u}} = \sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u})$$

Assumptions: $\text{diameter}(\mathcal{U}) \leq D$, $\|\boldsymbol{g}_t\|_2 \leq G$.

Standard Methods

Online Gradient Descent (OGD)

Move into direction of steepest descent:

$$\begin{aligned} \boldsymbol{w}_{t+1} &= \arg\min_{\boldsymbol{w}\in\mathcal{U}} \ \langle \boldsymbol{w}, \boldsymbol{g}_t \rangle + \frac{1}{2\eta_t} \|\boldsymbol{w} - \boldsymbol{w}_t\|_2^2 \\ &= \boldsymbol{w}_t - \eta_t \boldsymbol{g}_t \qquad \text{(and project onto } \mathcal{U} \text{ if go outside)} \end{aligned}$$

Regularization/step size determined by **learning rate** $\eta_t > 0$.

Standard Methods

Online Gradient Descent (OGD)

Move into direction of steepest descent:

$$\begin{aligned} \boldsymbol{w}_{t+1} &= \arg\min_{\boldsymbol{w}\in\mathcal{U}} \ \langle \boldsymbol{w}, \boldsymbol{g}_t \rangle + \frac{1}{2\eta_t} \|\boldsymbol{w} - \boldsymbol{w}_t\|_2^2 \\ &= \boldsymbol{w}_t - \eta_t \boldsymbol{g}_t \qquad \text{(and project onto } \mathcal{U} \text{ if go outside)} \end{aligned}$$

Regularization/step size determined by **learning rate** $\eta_t > 0$.

Online Newton Step (ONS)

Make less sensitive to parametrization by running OGD on pre-conditioned functions $\ell_t(\boldsymbol{H}_{t+1}^{-1/2}\tilde{\boldsymbol{w}})$:

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

where $\boldsymbol{H}_{t+1} = I + 2\sum_{s=1}^{t} \boldsymbol{g}_s \boldsymbol{g}_s^\intercal$ and η fixed.

The Standard Picture

Minimax rates based on curvature [Hazan, 2016]:

Convex ℓ_t	$\sqrt{T}$	GD with $\eta_t \propto \frac{1}{\sqrt{t}}$
Strongly convex ℓ_t	$\ln T$	GD with $\eta_t \propto \frac{1}{t}$
Exp-concave ℓ_t	$d \ln T$	ONS with $\eta \propto 1$

- **Strongly convex**: second derivative at least $\alpha > 0$, implies exp-concave
- **Exp-concave**: $e^{-\alpha \ell_t}$ concave
  Satisfied by logistic loss, squared loss, but not hinge loss

The Standard Picture

Minimax rates based on curvature [Hazan, 2016]:

Convex ℓ_t	$\sqrt{T}$	GD with $\eta_t \propto \frac{1}{\sqrt{t}}$
Strongly convex ℓ_t	$\ln T$	GD with $\eta_t \propto \frac{1}{t}$
Exp-concave ℓ_t	$d \ln T$	ONS with $\eta \propto 1$

Limitations:

- Different method in each case! (Requires sophisticated users.)
- Theoretical tuning of η_t **very conservative**
- What if curvature varies between rounds?
- In many applications data are **stochastic** (i.i.d.) Should be easier than worst case...

Need Adaptive Methods!

Existing Adaptivity Results

- [Bartlett, Hazan, and Rakhlin, 2007], [Do et al., 2009]: Adaptive GD: **strongly convex + general convex**

Other Types of Adaptivity:

- [Orabona, 2014, Orabona and Pál, 2016]: adapt to size $\|\boldsymbol{u}\|_2$ of comparator
- AdaGrad [Duchi et al., 2011]: box-like domain (ℓ_∞-ball) instead of ℓ_2-ball
- [Hazan and Kale, 2010], [Chiang et al., 2012]: linear functions ℓ_t that vary little over time
- [Orabona, Crammer, and Cesa-Bianchi, 2015]: data-dependent time-varying regularizers
- ...

Key techniques:

- Adaptive tuning of learning rate η_t

Main difficulty:

- Overhead for learning η_t dominates benefits of tuning η_t

MetaGrad: Multiple Eta Gradient Algorithm

Theorem (Van Erven, Koolen, 2016)

MetaGrad's $\text{Regret}_T^{\boldsymbol{u}}$ *is bounded by*

$$\sum_{t=1}^{T}(\boldsymbol{w}_t - \boldsymbol{u})^\mathsf{T}\boldsymbol{g}_t \preccurlyeq \begin{cases} \sqrt{T \ln\ln T} \\ \\ \sqrt{V_T^{\boldsymbol{u}}\, d \ln T} + d \ln T, \end{cases}$$

where

$$V_T^{\boldsymbol{u}} = \sum_{t=1}^{T}((\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t)^2$$

- By convexity, $\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u}) \leq (\boldsymbol{w}_t - \boldsymbol{u})^\mathsf{T}\boldsymbol{g}_t$.

MetaGrad: Multiple Eta Gradient Algorithm

Theorem (Van Erven, Koolen, 2016)

MetaGrad's $\text{Regret}_T^{\boldsymbol{u}}$ *is bounded by*

$$\sum_{t=1}^{T}(\boldsymbol{w}_t - \boldsymbol{u})^\mathsf{T}\boldsymbol{g}_t \preccurlyeq \begin{cases} \sqrt{T \ln\ln T} \\ \\ \sqrt{V_T^{\boldsymbol{u}}\, d \ln T} + d \ln T, \end{cases}$$

where

$$V_T^{\boldsymbol{u}} = \sum_{t=1}^{T}((\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t)^2 = \sum_{t=1}^{T}(\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t\boldsymbol{g}_t^\mathsf{T}(\boldsymbol{u} - \boldsymbol{w}_t).$$

- By convexity, $\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u}) \leq (\boldsymbol{w}_t - \boldsymbol{u})^\mathsf{T}\boldsymbol{g}_t$.
- Covariance: $\boldsymbol{g}_t\boldsymbol{g}_t^\mathsf{T} \propto \boldsymbol{X}_t\boldsymbol{X}_t^\mathsf{T}$ when $\ell_t(\boldsymbol{w}) = h_t(\langle \boldsymbol{w}, \boldsymbol{X}_t \rangle)$
  e.g. hinge, logistic, squared loss

MetaGrad: Multiple Eta Gradient Algorithm

Theorem (Van Erven, Koolen, 2016)

MetaGrad's $\text{Regret}_T^{\boldsymbol{u}}$ *is bounded by*

$$\sum_{t=1}^{T}(\boldsymbol{w}_t - \boldsymbol{u})^\mathsf{T}\boldsymbol{g}_t \preccurlyeq \begin{cases} \sqrt{T \ln\ln T} \\ \\ \sqrt{V_T^{\boldsymbol{u}}\, d \ln T} + d \ln T, \end{cases}$$

where

$$V_T^{\boldsymbol{u}} = \sum_{t=1}^{T}((\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t)^2 = \sum_{t=1}^{T}(\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t\boldsymbol{g}_t^\mathsf{T}(\boldsymbol{u} - \boldsymbol{w}_t).$$

- By convexity, $\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u}) \leq (\boldsymbol{w}_t - \boldsymbol{u})^\mathsf{T}\boldsymbol{g}_t$.
- Covariance: $\boldsymbol{g}_t\boldsymbol{g}_t^\mathsf{T} \propto \boldsymbol{X}_t\boldsymbol{X}_t^\mathsf{T}$ when $\ell_t(\boldsymbol{w}) = h_t(\langle \boldsymbol{w}, \boldsymbol{X}_t \rangle)$
  e.g. hinge, logistic, squared loss
- Optimal learning rate η depends on $V_T^{\boldsymbol{u}}$, but $\boldsymbol{u}$ unknown!
  Solution: aggregate **multiple learning rates** at almost no cost

Consequences

1. Non-stochastic adaptation to curvature:

Convex ℓ_t	$\sqrt{T \ln \ln T}$
Exp-concave ℓ_t	$d \ln T$
Fixed convex $\ell_t = \ell$	$d \ln T$

Consequences

1. Non-stochastic adaptation to curvature:

Convex ℓ_t	$\sqrt{T \ln \ln T}$
Exp-concave ℓ_t	$d \ln T$
Fixed convex $\ell_t = \ell$	$d \ln T$

Loose end: strongly convex $\Rightarrow$ exp-concave gives $d \ln T$

Consequences

1. Non-stochastic adaptation to curvature:

Convex ℓ_t	$\sqrt{T \ln \ln T}$
Exp-concave ℓ_t	$d \ln T$
Fixed convex $\ell_t = \ell$	$d \ln T$

Loose end: strongly convex $\Rightarrow$ exp-concave gives $d \ln T$

2. Stochastic without curvature

Suppose ℓ_t i.i.d. with stochastic optimum $\boldsymbol{u}^* = \arg\min_{\boldsymbol{u} \in \mathcal{U}} \mathbb{E}_\ell[\ell(\boldsymbol{u})]$. Then expected regret $\mathbb{E}[\text{Regret}_T^{\boldsymbol{u}^*}]$:

Absolute loss* $\ell_t(u) = \lvert u - X_t \rvert$	$\ln T$
Hinge loss* $\max\{0, 1 - Y_t \langle \boldsymbol{u}, \boldsymbol{X}_t \rangle\}$	$d \ln T$
(B, β)-**Bernstein**	$(Bd \ln T)^{1/(2-\beta)} \, T^{(1-\beta)/(2-\beta)}$

*Conditions apply

1. Directional Derivative Condition

Corollary (Van Erven, Koolen, 2016)

If there exist $a, b > 0$ such that all ℓ_t satisfy

$$\ell_t(\boldsymbol{u}) \geq \ell_t(\boldsymbol{w}) + a(\boldsymbol{u} - \boldsymbol{w})^\intercal \nabla \ell_t(\boldsymbol{w}) + b\left((\boldsymbol{u} - \boldsymbol{w})^\intercal \nabla \ell_t(\boldsymbol{w})\right)^2$$

for all $\boldsymbol{w} \in \mathcal{U}$, then $O(d \ln T)$ regret w.r.t. $\boldsymbol{u}$.

$a = 1$

- Satisfied by **exp-concave** functions [Hazan et al., 2007]
- Requires quadratic curvature in direction of minimizer $\boldsymbol{u}$.

General a

- Satisfied for any **fixed convex** function $\ell_t = f$ with minimizer $\boldsymbol{u}$, even without any curvature, with $a = 2$ and $b = 1/(DG)$.

2. Bernstein Condition for Online Learning

Suppose ℓ_t i.i.d. with stochastic optimum $\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}\in\mathcal{U}} \mathop{\mathbb{E}}_{\ell}[\ell(\boldsymbol{u})]$.

Standard Bernstein condition:

$$\mathbb{E}\left(\ell(\boldsymbol{w}) - \ell(\boldsymbol{u}^*)\right)^2 \leq B\Big(\mathbb{E}\left[\ell(\boldsymbol{w}) - \ell(\boldsymbol{u}^*)\right]\Big)^{\beta} \qquad \text{for all } \boldsymbol{w} \in \mathcal{U}.$$

2. Bernstein Condition for Online Learning

Suppose ℓ_t i.i.d. with stochastic optimum $\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}\in\mathcal{U}} \mathop{\mathbb{E}}_{\ell}[\ell(\boldsymbol{u})]$.

Standard Bernstein condition:

$$\mathbb{E}\left(\ell(\boldsymbol{w}) - \ell(\boldsymbol{u}^*)\right)^2 \leq B\Big(\mathbb{E}\left[\ell(\boldsymbol{w}) - \ell(\boldsymbol{u}^*)\right]\Big)^{\beta} \qquad \text{for all } \boldsymbol{w} \in \mathcal{U}.$$

Replace by **weaker linearized version:**

- Apply with $\tilde{\ell}(\boldsymbol{u}) = \langle \boldsymbol{u}, \nabla\,\ell(\boldsymbol{w})\rangle$ instead of ℓ!
- By convexity, $\ell(\boldsymbol{w}) - \ell(\boldsymbol{u}^*) \leq \tilde{\ell}(\boldsymbol{w}) - \tilde{\ell}(\boldsymbol{u}^*)$.

$$\mathbb{E}\left((\boldsymbol{w} - \boldsymbol{u}^*)\nabla\,\ell(\boldsymbol{w})\right)^2 \leq B\Big(\mathbb{E}\left[(\boldsymbol{w} - \boldsymbol{u}^*)\nabla\,\ell(\boldsymbol{w})\right]\Big)^{\beta} \qquad \text{for all } \boldsymbol{w} \in \mathcal{U}.$$

2. Bernstein Condition for Online Learning

Suppose ℓ_t i.i.d. with stochastic optimum $\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}\in\mathcal{U}} \mathbb{E}_{\ell}[\ell(\boldsymbol{u})]$.

Standard Bernstein condition:

$$\mathbb{E}\,(\ell(\boldsymbol{w}) - \ell(\boldsymbol{u}^*))^2 \le B\big(\,\mathbb{E}\,[\ell(\boldsymbol{w}) - \ell(\boldsymbol{u}^*)]\,\big)^{\beta} \qquad \text{for all } \boldsymbol{w}\in\mathcal{U}.$$

Replace by **weaker linearized version:**

- Apply with $\tilde{\ell}(\boldsymbol{u}) = \langle \boldsymbol{u}, \nabla\,\ell(\boldsymbol{w})\rangle$ instead of ℓ!
- By convexity, $\ell(\boldsymbol{w}) - \ell(\boldsymbol{u}^*) \le \tilde{\ell}(\boldsymbol{w}) - \tilde{\ell}(\boldsymbol{u}^*)$.

$$\mathbb{E}\,((\boldsymbol{w} - \boldsymbol{u}^*)\nabla\,\ell(\boldsymbol{w}))^2 \le B\big(\,\mathbb{E}\,[(\boldsymbol{w} - \boldsymbol{u}^*)\nabla\,\ell(\boldsymbol{w})]\,\big)^{\beta} \qquad \text{for all } \boldsymbol{w}\in\mathcal{U}.$$

Hinge loss (with $G = D = 1$): $\beta = 1$, $B = \frac{2\lambda_{\max}(\mathbb{E}[\boldsymbol{X}\boldsymbol{X}^{\mathsf{T}}])}{\|\,\mathbb{E}[Y\boldsymbol{X}]\|}$

2. Bernstein Condition for Online Learning

Suppose ℓ_t i.i.d. with stochastic optimum $\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}\in\mathcal{U}} \mathop{\mathbb{E}}_{\ell}[\ell(\boldsymbol{u})]$.

Standard Bernstein condition:

$$\mathbb{E}\left(\ell(\boldsymbol{w}) - \ell(\boldsymbol{u}^*)\right)^2 \leq B\Big(\mathbb{E}\left[\ell(\boldsymbol{w}) - \ell(\boldsymbol{u}^*)\right]\Big)^{\beta} \qquad \text{for all } \boldsymbol{w} \in \mathcal{U}.$$

Replace by **weaker linearized version:**

- Apply with $\tilde{\ell}(\boldsymbol{u}) = \langle \boldsymbol{u}, \nabla \ell(\boldsymbol{w})\rangle$ instead of ℓ!
- By convexity, $\ell(\boldsymbol{w}) - \ell(\boldsymbol{u}^*) \leq \tilde{\ell}(\boldsymbol{w}) - \tilde{\ell}(\boldsymbol{u}^*)$.

$$\mathbb{E}\left((\boldsymbol{w} - \boldsymbol{u}^*)\nabla \ell(\boldsymbol{w})\right)^2 \leq B\Big(\mathbb{E}\left[(\boldsymbol{w} - \boldsymbol{u}^*)\nabla \ell(\boldsymbol{w})\right]\Big)^{\beta} \qquad \text{for all } \boldsymbol{w} \in \mathcal{U}.$$

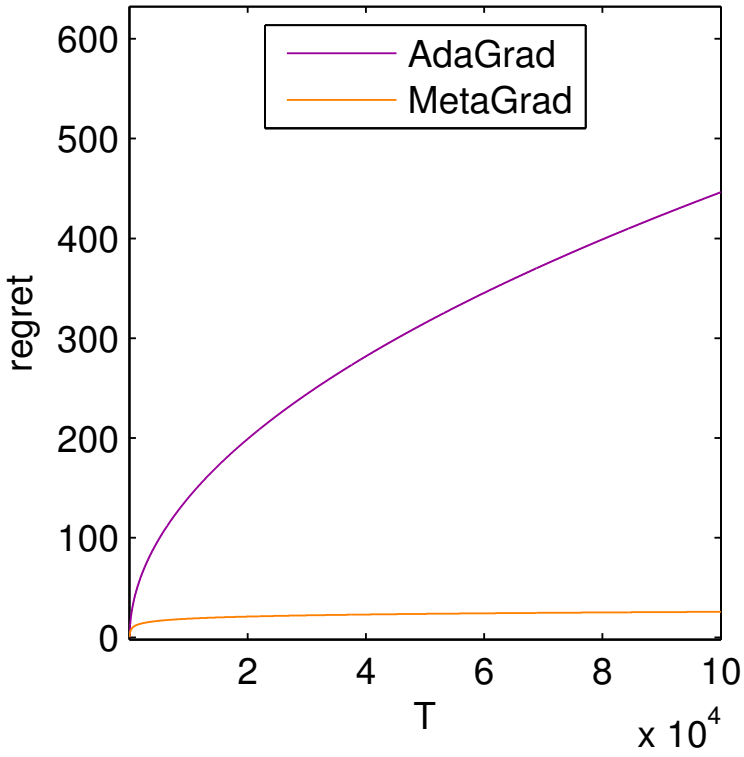
Hinge loss (with $G = D = 1$): $\beta = 1$, $B = \frac{2\lambda_{\max}(\mathbb{E}[\boldsymbol{X}\boldsymbol{X}^{\mathsf{T}}])}{\|\mathbb{E}[Y\boldsymbol{X}]\|}$

Theorem (Koolen, Grünwald, Van Erven, 2016)

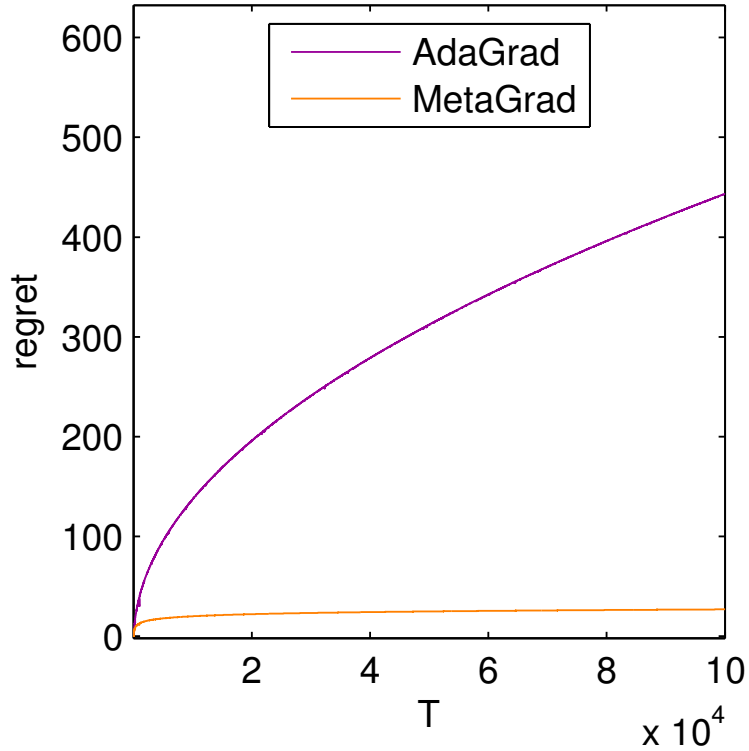
$$\mathbb{E}[\mathrm{Regret}_T^{\boldsymbol{u}^*}] \preccurlyeq (Bd \ln T)^{1/(2-\beta)}\, T^{(1-\beta)/(2-\beta)}$$

$$\mathrm{Regret}_T^{\boldsymbol{u}^*} \preccurlyeq (Bd \ln T - \ln \delta)^{1/(2-\beta)}\, T^{(1-\beta)/(2-\beta)} \quad \textit{w.p.} \geq 1-\delta$$

Difference in Rates Not Just Theoretical



Offline: $\ell_t(u) = |u - 1/4|$



Stochastic Online: $\ell_t(u) = |u - X_t|$
where $X_t = \pm\frac{1}{2}$ i.i.d. w.p. 0.4 and 0.6.

- MetaGrad: $O(\ln T)$ regret, AdaGrad: $O(\sqrt{T})$, match bounds
- Functions neither strongly convex nor smooth
- **Caveat:** comparison more complicated for higher dimensions, unless we run a separate copy of MetaGrad per dimension, like the diagonal version of AdaGrad runs GD per dimension

MetaGrad Algorithm

Second-order **surrogate loss** for each η of interest (from a grid):

$$\ell_t^\eta(\boldsymbol{u}) = \eta(\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t + \eta^2(\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t\boldsymbol{g}_t^\mathsf{T}(\boldsymbol{u} - \boldsymbol{w}_t)$$

MetaGrad Algorithm

Second-order **surrogate loss** for each η of interest (from a grid):

$$\ell_t^\eta(\boldsymbol{u}) = \eta(\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t + \eta^2(\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t\boldsymbol{g}_t^\mathsf{T}(\boldsymbol{u} - \boldsymbol{w}_t)$$

One **Slave** algorithm per η produces $\boldsymbol{w}_t^\eta$ such that

$$\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta) - \sum_{t=1}^T \ell_t^\eta(\boldsymbol{u}) \leq R_{\text{slave}}^{\boldsymbol{u}}(\eta)$$

MetaGrad Algorithm

Second-order **surrogate loss** for each η of interest (from a grid):

$$\ell_t^\eta(\boldsymbol{u}) = \eta(\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t + \eta^2(\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t\boldsymbol{g}_t^\mathsf{T}(\boldsymbol{u} - \boldsymbol{w}_t)$$

One **Slave** algorithm per η produces $\boldsymbol{w}_t^\eta$ such that

$$\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta) - \sum_{t=1}^T \ell_t^\eta(\boldsymbol{u}) \leq R_{\text{slave}}^{\boldsymbol{u}}(\eta)$$

Single **Master** algorithm produces $\boldsymbol{w}_t$ such that

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

MetaGrad Algorithm

Second-order **surrogate loss** for each η of interest (from a grid):

$$\ell_t^\eta(\boldsymbol{u}) = \eta(\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t + \eta^2(\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t\boldsymbol{g}_t^\mathsf{T}(\boldsymbol{u} - \boldsymbol{w}_t)$$

One **Slave** algorithm per η produces $\boldsymbol{w}_t^\eta$ such that

$$\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta) - \sum_{t=1}^T \ell_t^\eta(\boldsymbol{u}) \leq R_{\text{slave}}^{\boldsymbol{u}}(\eta)$$

Single **Master** algorithm produces $\boldsymbol{w}_t$ such that

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

Together: $-\sum_{t=1}^T \ell_t^\eta(\boldsymbol{u}) \leq R_{\text{slave}}^{\boldsymbol{u}}(\eta) + R_{\text{master}}(\eta) \quad \forall \eta$

MetaGrad Algorithm

Second-order **surrogate loss** for each η of interest (from a grid):

$$\ell_t^\eta(\boldsymbol{u}) = \eta(\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t + \eta^2(\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t\boldsymbol{g}_t^\mathsf{T}(\boldsymbol{u} - \boldsymbol{w}_t)$$

One **Slave** algorithm per η produces $\boldsymbol{w}_t^\eta$ such that

$$\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta) - \sum_{t=1}^T \ell_t^\eta(\boldsymbol{u}) \leq R_{\text{slave}}^{\boldsymbol{u}}(\eta)$$

Single **Master** algorithm produces $\boldsymbol{w}_t$ such that

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

Together: $-\sum_{t=1}^T \ell_t^\eta(\boldsymbol{u}) \leq R_{\text{slave}}^{\boldsymbol{u}}(\eta) + R_{\text{master}}(\eta) \quad \forall \eta$

$$\sum_{t=1}^T (\boldsymbol{w}_t - \boldsymbol{u})^\mathsf{T}\boldsymbol{g}_t \leq \frac{R_{\text{slave}}^{\boldsymbol{u}}(\eta) + R_{\text{master}}(\eta)}{\eta} + \eta V_T^{\boldsymbol{u}}$$

MetaGrad Algorithm

Second-order **surrogate loss** for each η of interest (from a grid):

$$\ell_t^\eta(\boldsymbol{u}) = \eta(\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t + \eta^2(\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t\boldsymbol{g}_t^\mathsf{T}(\boldsymbol{u} - \boldsymbol{w}_t)$$

One **Slave** algorithm per η produces $\boldsymbol{w}_t^\eta$ such that

$$\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta) - \sum_{t=1}^T \ell_t^\eta(\boldsymbol{u}) \leq R_{\text{slave}}^{\boldsymbol{u}}(\eta)$$

Single **Master** algorithm produces $\boldsymbol{w}_t$ such that

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

Together: $-\sum_{t=1}^T \ell_t^\eta(\boldsymbol{u}) \leq R_{\text{slave}}^{\boldsymbol{u}}(\eta) + R_{\text{master}}(\eta) \quad \forall \eta$

$$\sum_{t=1}^T (\boldsymbol{w}_t - \boldsymbol{u})^\mathsf{T}\boldsymbol{g}_t \leq \frac{O(d \ln T) + O(\ln \ln T)}{\eta} + \eta V_T^{\boldsymbol{u}}$$

MetaGrad Algorithm

Second-order **surrogate loss** for each η of interest (from a grid):

$$\ell_t^\eta(\boldsymbol{u}) = \eta(\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t + \eta^2(\boldsymbol{u} - \boldsymbol{w}_t)^\mathsf{T}\boldsymbol{g}_t\boldsymbol{g}_t^\mathsf{T}(\boldsymbol{u} - \boldsymbol{w}_t)$$

One **Slave** algorithm per η produces $\boldsymbol{w}_t^\eta$ such that

$$\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta) - \sum_{t=1}^T \ell_t^\eta(\boldsymbol{u}) \leq R_{\text{slave}}^{\boldsymbol{u}}(\eta)$$

Single **Master** algorithm produces $\boldsymbol{w}_t$ such that

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

Together: $-\sum_{t=1}^T \ell_t^\eta(\boldsymbol{u}) \leq R_{\text{slave}}^{\boldsymbol{u}}(\eta) + R_{\text{master}}(\eta) \quad \forall \eta$

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

MetaGrad Master

Goal: aggregate slave predictions $\boldsymbol{w}_t^\eta$ for all η in exponentially spaced grid $\frac{2^{-0}}{5DG}, \frac{2^{-1}}{5DG}, \ldots, \frac{2^{-\lceil \frac{1}{2}\log_2 T \rceil}}{5DG}$

Difficulty: master's predictions must be good w.r.t. different loss functions ℓ_t^η for all η simultaneously

Compute **exponential weights** with performance of each η measured by its own surrogate loss:

$$\pi_t(\eta) = \frac{\pi_1(\eta) e^{-\sum_{s<t} \ell_s^\eta(\boldsymbol{w}_s^\eta)}}{Z}$$

Then predict with **tilted** exponentially weighted average:

$$\boldsymbol{w}_t = \frac{\sum_\eta \pi_t(\eta)\, \eta\, \boldsymbol{w}_t^\eta}{\sum_\eta \pi_t(\eta)\, \eta}$$

MetaGrad Master Analysis

Potential $\Phi_T = \sum_\eta \pi_1(\eta) e^{-\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta)}$

Proof outline:

$$\Phi_T \leq \Phi_{T-1} \leq \cdots \leq \Phi_0 = 1$$

$$\pi_1(\eta) e^{-\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta)} \leq 1 \qquad \forall \eta$$

$$\underbrace{\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t)}_{=0} - \sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta) \leq -\ln \pi_1(\eta)$$

MetaGrad Master Analysis

Potential $\Phi_T = \sum_\eta \pi_1(\eta) e^{-\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta)}$

Proof outline:

$$\Phi_T \leq \Phi_{T-1} \leq \cdots \leq \Phi_0 = 1$$

$$\pi_1(\eta) e^{-\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta)} \leq 1 \qquad \forall \eta$$

$$\underbrace{\sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t)}_{=0} - \sum_{t=1}^T \ell_t^\eta(\boldsymbol{w}_t^\eta) \leq -\ln \pi_1(\eta)$$

Grid has $\lceil \frac{1}{2} \log_2 T \rceil + 1$ learning rates, so for heavy-tailed prior:

$$-\ln \pi_1(\eta) = O(\ln \ln T)$$

MetaGrad Master Analysis: Decreasing Potential

Surrogate loss $\ell_t^\eta(\boldsymbol{u}) = \eta(\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \boldsymbol{g}_t + \eta^2(\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \boldsymbol{g}_t \boldsymbol{g}_t^\intercal (\boldsymbol{u} - \boldsymbol{w}_t)$ is **exp-concave**, even if ℓ_t is not.

Upper bound by tangent at $\boldsymbol{u} = \boldsymbol{w}_t$:

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

MetaGrad Master Analysis: Decreasing Potential

Surrogate loss $\ell_t^\eta(\boldsymbol{u}) = \eta(\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \boldsymbol{g}_t + \eta^2(\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \boldsymbol{g}_t \boldsymbol{g}_t^\intercal (\boldsymbol{u} - \boldsymbol{w}_t)$ is **exp-concave**, even if ℓ_t is not.

Upper bound by tangent at $\boldsymbol{u} = \boldsymbol{w}_t$:

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

Choose master's weights to ensure decreasing potential:

$$\begin{aligned}
\Phi_T - \Phi_{T-1} &= \sum_\eta \pi_1(\eta) e^{-\sum_{t<T} \ell_t^\eta(\boldsymbol{w}_t^\eta)} \left(e^{-\ell_T^\eta(\boldsymbol{w}_T^\eta)} - 1 \right) \\
&\leq \sum_\eta \pi_1(\eta) e^{-\sum_{t<T} \ell_t^\eta(\boldsymbol{w}_t^\eta)} \eta(\boldsymbol{w}_T - \boldsymbol{w}_T^\eta)^\intercal \boldsymbol{g}_T \\
&= 0 \qquad \text{for any } \boldsymbol{g}_T
\end{aligned}$$

MetaGrad Slave

Goal: Given η, minimize regret w.r.t. exp-concave surrogate ℓ_t^η.

Update:

$$\tilde{\boldsymbol{w}}_{t+1}^\eta = \boldsymbol{w}_t^\eta - \eta s_t^\eta \Sigma_{t+1}^\eta \boldsymbol{g}_t,$$

where

$$\Sigma_{t+1}^\eta = \left(\frac{1}{D^2} \boldsymbol{I} + 2\eta^2 \sum_{s=1}^{t} \boldsymbol{g}_s \boldsymbol{g}_s^\intercal \right)^{-1} \qquad s_t^\eta = 1 + 2\eta \boldsymbol{g}_t^\intercal (\boldsymbol{w}_t^\eta - \boldsymbol{w}_t)$$

Project onto domain:

$$\boldsymbol{w}_{t+1}^\eta = \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathcal{U}} \quad (\boldsymbol{u} - \tilde{\boldsymbol{w}}_{t+1}^\eta)^\intercal (\Sigma_{t+1}^\eta)^{-1} (\boldsymbol{u} - \tilde{\boldsymbol{w}}_{t+1}^\eta)$$

MetaGrad Slave

Goal: Given η, minimize regret w.r.t. exp-concave surrogate ℓ_t^η.

Update:

$$\tilde{\boldsymbol{w}}_{t+1}^\eta = \boldsymbol{w}_t^\eta - \eta s_t^\eta \Sigma_{t+1}^\eta \boldsymbol{g}_t,$$

where

$$\Sigma_{t+1}^\eta = \left(\frac{1}{D^2} \boldsymbol{I} + 2\eta^2 \sum_{s=1}^{t} \boldsymbol{g}_s \boldsymbol{g}_s^\intercal \right)^{-1} \qquad s_t^\eta = 1 + 2\eta \boldsymbol{g}_t^\intercal (\boldsymbol{w}_t^\eta - \boldsymbol{w}_t)$$

Project onto domain:

$$\boldsymbol{w}_{t+1}^\eta = \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathcal{U}} \quad (\boldsymbol{u} - \tilde{\boldsymbol{w}}_{t+1}^\eta)^\intercal (\Sigma_{t+1}^\eta)^{-1} (\boldsymbol{u} - \tilde{\boldsymbol{w}}_{t+1}^\eta)$$

- If master = slave, i.e. $\boldsymbol{w}_t^\eta = \boldsymbol{w}_t$, then is **Online Newton Step**

Summary

1. Statistical learning

- Limits of minimax analysis pretty well understood
- Margin/Bernstein condition relates variance to excess risk
- Adaptive algorithms known to exploit 'easy data'

2. Online sequential prediction

- Most results still about minimax analysis
- MetaGrad:
  - new adaptive algorithm
  - with new variance bound
- Variance bound implies fast rates in:
  - all known cases: exp-concave, strong convex
  - new cases characterized by new online version of Bernstein condition

Future Work

Computation

- Online learning often applied in high dimensions d
- Gradient descent: $O(d)$ work per step
- MetaGrad: $O(d^2)$ work per step + projection on domain
- Need to speed up MetaGrad to work in high dimensions

Important Application: Deep Learning

- Online learning (e.g. gradient descent) is used to train large neural networks
- But loss is not convex, so theory breaks down
- Make it work experimentally

Papers

- T. van Erven and W. M. Koolen. **Metagrad: Multiple learning rates in online learning.** In Advances in Neural Information Processing Systems 29 (NIPS), pages 3666–3674, 2016.
- W. M. Koolen, P. Grünwald, and T. van Erven. **Combining adversarial guarantees and stochastic fast rates in online learning.** In Advances in Neural Information Processing Systems 29 (NIPS), pages 4457–4465, 2016.

References

P. L. Bartlett, E. Hazan, and A. Rakhlin. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems 20 (NIPS)*, pages 65–72, 2007.

C.-K. Chiang, T. Yang, C.-J. Le, M. Mahdavi, C.-J. Lu, R. Jin, and S. Zhu. Online optimization with gradual variations. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, pages 6.1–6.20, 2012.

C. B. Do, Q. V. Le, and C.-S. Foo. Proximal regularization for online and batch learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 257–264, 2009.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

E. Hazan. Introduction to online optimization. Draft, April 10, 2016, available from `ocobook.cs.princeton.edu`, 2016.

E. Hazan and S. Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. *Machine learning*, 80(2-3):165–188, 2010.

E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

H. Luo, A. Agarwal, N. Cesa-Bianchi, and J. Langford. Efficient second order online learning by sketching. In *Advances in Neural Information Processing Systems 29 (NIPS)*, 2016.

F. Orabona. Simultaneous model selection and optimization through parameter-free stochastic learning. In *NIPS 27*, pages 1116–1124, 2014.

F. Orabona and D. Pál. Coin betting and parameter-free online learning. In *NIPS 29*, 2016.

F. Orabona, K. Crammer, and N. Cesa-Bianchi. A generalized online mirror descent with applications to classification and regression. *Machine Learning*, 99(3):411–435, 2015.

A. B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.