MetaGrad: Multiple Learning Rates in Online Learning

Tim van Erven  Wouter Koolen
Online Convex Optimization

Parameters $w$ take values in a convex domain $U$

1: for $t = 1, 2, \ldots, T$ do
2: Learner plays $w_t \in U$
3: Environment reveals convex loss function $f_t : U \rightarrow \mathbb{R}$
4: Learner incurs loss $f_t(w_t)$, observes gradient $g_t = \nabla f_t(w_t)$
5: end for

Measure regret w.r.t. $u \in U$:

$$\text{Regret}_T^u = \sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(u)$$

Assumptions: $\text{diameter}(U) \leq D$, $\|g_t\|_2 \leq G$. 
The Standard Picture

Rates based on curvature:

<table>
<thead>
<tr>
<th>Convex $f_t$</th>
<th>$\sqrt{T}$</th>
<th>GD with $\eta_t \propto \frac{1}{\sqrt{t}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly convex $f_t$</td>
<td>$\ln T$</td>
<td>GD with $\eta_t \propto \frac{1}{t}$</td>
</tr>
<tr>
<td>Exp-concave $f_t$</td>
<td>$d \ln T$</td>
<td>ONS with $\eta_t = \text{constant}$</td>
</tr>
</tbody>
</table>

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

Rates based on curvature:

<table>
<thead>
<tr>
<th>Function Type</th>
<th>Rate</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convex $f_t$</td>
<td>$\sqrt{T}$</td>
<td>GD with $\eta_t \propto \frac{1}{\sqrt{t}}$</td>
</tr>
<tr>
<td>Strongly convex $f_t$</td>
<td>$\ln T$</td>
<td>GD with $\eta_t \propto \frac{1}{t}$</td>
</tr>
<tr>
<td>Exp-concave $f_t$</td>
<td>$d \ln T$</td>
<td>ONS with $\eta_t = \text{constant}$</td>
</tr>
</tbody>
</table>

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

Our goals:

- Adaptivity to more types of functions $f_t$
- Fast rates without curvature for ‘easy’ stochastic data
Other Types of Adaptivity

- [Orabona, 2014, Orabona and Pál, 2016]: adapt to size $\|u\|_2$ of comparator
- AdaGrad [Duchi et al., 2011]: box-like domain ($\ell_\infty$-ball) instead of $\ell_2$-ball
- [Hazan and Kale, 2010], [Chiang et al., 2012]: linear functions $f_t$ that vary little over time
- [Orabona, Crammer, and Cesa-Bianchi, 2015]: data-dependent time-varying regularizers

Key techniques:

- Adaptive tuning of learning rate $\eta_t$
- Use second-order information about covariance of features in time-varying regularizer
MetaGrad’s Regret is bounded by

\[ \sum_{t=1}^{T} (w_t - u)^\top g_t = \begin{cases} O \left( \sqrt{V_u^T} d \ln T + d \ln T \right) \\ O(\sqrt{T \ln \ln T}), \end{cases} \]

where

\[ V_u^T = \sum_{t=1}^{T} (u - w_t)^\top g_t g_t^\top (u - w_t). \]

- By convexity, \( f_t(w_t) - f_t(u) \leq (w_t - u)^\top g_t. \)
MetaGrad's Regret $\sum_{t=1}^{T} (w_t - u)^T g_t$ is bounded by

\[
\begin{cases}
O\left(\sqrt{V_T^u} d \ln T + d \ln T\right), \\
O(\sqrt{T \ln \ln T}),
\end{cases}
\]

where

\[
V_T^u = \sum_{t=1}^{T} (u - w_t)^T g_t g_t^T (u - w_t).
\]

- By convexity, $f_t(w_t) - f_t(u) \leq (w_t - u)^T g_t$.
- Covariance: $g_t g_t^T \propto X_t X_t^T$ when $f_t(w) = \text{loss}(Y_t \langle w, X_t \rangle)$.
MetaGrad: Multiple Eta Gradient Algorithm

**Theorem**

MetaGrad’s Regret $\sum_{t=1}^{T} (w_t - u)^T g_t$ is bounded by

$$\sum_{t=1}^{T} (w_t - u)^T g_t = \begin{cases} O\left(\sqrt{V_T^u} d \ln T + d \ln T\right) \\ O\left(\sqrt{T \ln \ln T}\right), \end{cases}$$

where

$$V_T^u = \sum_{t=1}^{T} (u - w_t)^T g_t g_t^T (u - w_t).$$

- By convexity, $f_t(w_t) - f_t(u) \leq (w_t - u)^T g_t$.
- Covariance: $g_t g_t^T \propto X_t X_t^T$ when $f_t(w) = \text{loss}(Y_t \langle w, X_t \rangle)$.
- Optimal learning rate depends on $V_T^u$, but $u$ unknown! Solution: aggregate multiple learning rates.
Consequences

Non-stochastic adaptation:

<table>
<thead>
<tr>
<th>Convex $f_t$</th>
<th>$\sqrt{T \ln \ln T}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp-concave $f_t$</td>
<td>$d \ln T$</td>
</tr>
<tr>
<td>Fixed convex $f_t = f$</td>
<td>$d \ln T$</td>
</tr>
</tbody>
</table>
Consequences

Non-stochastic adaptation:

<table>
<thead>
<tr>
<th>Convex $f_t$</th>
<th>$\sqrt{T \ln \ln T}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp-concave $f_t$</td>
<td>$d \ln T$</td>
</tr>
<tr>
<td>Fixed convex $f_t = f$</td>
<td>$d \ln T$</td>
</tr>
</tbody>
</table>

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

Non-stochastic adaptation:

<table>
<thead>
<tr>
<th>Convex $f_t$</th>
<th>$\sqrt{T \ln \ln T}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp-concave $f_t$</td>
<td>$d \ln T$</td>
</tr>
<tr>
<td>Fixed convex $f_t = f$</td>
<td>$d \ln T$</td>
</tr>
</tbody>
</table>

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

Stochastic without curvature [joint work with Grünwald]

Suppose $f_t$ i.i.d. with stochastic optimum $u^* = \arg \min_{u \in U} \mathbb{E}_f[f(u)]$. Then expected regret $\mathbb{E}[\text{Regret}_{u^*}^T]$

| Absolute loss* $f_t(u) = |u - X_t|$ | $\ln T$ |
|-------------------------------------|----------|
| Hinge loss* $\max\{0, 1 - Y_t\langle u, X_t \rangle\}$ | $d \ln T$ |
| $(B, \beta)$-Bernstein             | $(Bd \ln T)^{1/(2-\beta)} T^{(1-\beta)/(2-\beta)}$ |

*Conditions apply
Outline

Two General Fast Rate Conditions

MetaGrad Algorithm

Exponential Weights Interpretation of Online Learning Algorithms
1. Directional Derivative Condition

**Theorem**

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

$$f_t(u) \geq f_t(w) + a(u - w)^\top \nabla f_t(w) + b ((u - w)^\top \nabla f_t(w))^2$$

*for $w \in U$,*

*then $O(d \ln T)$ regret w.r.t. $u$.*

$a = 1$

- Satisfied by **exp-concave** functions [Hazan, Agarwal, and Kale, 2007]
- Requires quadratic curvature in direction of minimizer $u$.

**General $a$**

- Satisfied for any **fixed convex** function $f_t = f$ with minimizer $u$, even without any curvature, with $a = 2$ and $b = 1/(DG)$. 
2. Bernstein Condition for Online Learning

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

**Standard Bernstein condition:**

$$\mathbb{E} (f(w) - f(u^*))^2 \leq B (\mathbb{E} [f(w) - f(u^*)])^\beta$$

for all $w \in \mathcal{U}$.
2. Bernstein Condition for Online Learning

Suppose $f_t$ i.i.d. with stochastic optimum $u^* = \arg\min_{u \in U} \mathbb{E}[f(u)]$.

Standard Bernstein condition:

$$\mathbb{E} \left( (f(w) - f(u^*))^2 \right) \leq B \left( \mathbb{E} [f(w) - f(u^*)] \right)^\beta$$

for all $w \in \mathcal{U}$.

Replace by weaker linearized version:

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

$$\mathbb{E} \left( (w - u^*) \nabla f(w) \right)^2 \leq B \left( \mathbb{E} [(w - u^*) \nabla f(w)] \right)^\beta$$

for all $w \in \mathcal{U}$.
2. Bernstein Condition for Online Learning

Suppose $f_t$ i.i.d. with stochastic optimum $u^* = \arg \min_{u \in U} \mathbb{E}[f(u)]$.

**Standard Bernstein condition:**

$$
\mathbb{E}(f(w) - f(u^*))^2 \leq B \left( \mathbb{E}[f(w) - f(u^*)] \right)^\beta \quad \text{for all } w \in U.
$$

Replace by **weaker linearized version:**

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

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

Hinge loss (with $G = D = 1$): $\beta = 1$, $B = \frac{2\lambda_{\text{max}}(\mathbb{E}[XX^T])}{\|\mathbb{E}[YX]\|}$
2. Bernstein Condition for Online Learning

Suppose $f_t$ i.i.d. with stochastic optimum $u^* = \arg\min_{u \in U} \mathbb{E}[f(u)]$.

**Standard Bernstein condition:**

$$\mathbb{E}(f(w) - f(u^*))^2 \leq B \left( \mathbb{E}[f(w) - f(u^*)] \right)^\beta \quad \text{for all } w \in U.$$ 

Replace by **weaker linearized version:**

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

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

**Hinge loss (with } G = D = 1): \beta = 1, B = \frac{2\lambda_{\max}(\mathbb{E}[XX^\top])}{\| \mathbb{E}[YX] \|} \)**

**Theorem (Koolen, Grünwald, Van Erven, 2016)**

$$\mathbb{E}[\text{Regret}_{u^*}^T] = O\left( (Bd \ln T)^{1/(2-\beta)} T^{(1-\beta)/(2-\beta)} \right)$$

$$\text{Regret}_{u^*}^T = O\left( (Bd \ln T - \ln \delta)^{1/(2-\beta)} T^{(1-\beta)/(2-\beta)} \right) \quad \text{w.p. } \geq 1 - \delta$$
Difference in Rates Not Just Theoretical

- 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
Outline

Two General Fast Rate Conditions

MetaGrad Algorithm

Exponential Weights Interpretation of Online Learning Algorithms
MetaGrad Algorithm

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

$$
\ell_t^\eta(u) = \eta(u - w_t)^T g_t + \eta^2 (u - w_t)^T g_t g_t^T (u - w_t)
$$
MetaGrad Algorithm

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

$$\ell^\eta_t(u) = \eta(u - w_t)^T g_t + \eta^2(u - w_t)^T g_t g_t^T (u - w_t)$$

One **Slave** algorithm per $\eta$ produces $w^\eta_t$ such that

$$\sum_{t=1}^{T} \ell^\eta_t(w^\eta_t) - \sum_{t=1}^{T} \ell^\eta_t(u) \leq R^u_{\text{slave}}(\eta)$$
MetaGrad Algorithm

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

$$\ell^\eta_t(u) = \eta(u - w_t)^T g_t + \eta^2(u - w_t)^T g_t g_t^T (u - w_t)$$

One **Slave** algorithm per $\eta$ produces $w^\eta_t$ such that

$$\sum_{t=1}^T \ell^\eta_t(w^\eta_t) - \sum_{t=1}^T \ell^\eta_t(u) \leq R^u_{\text{slave}}(\eta)$$

Single **Master** algorithm produces $w_t$ such that

$$\sum_{t=1}^T \ell^\eta_t(w_t) - \sum_{t=1}^T \ell^\eta_t(w^\eta_t) \leq R^u_{\text{master}}(\eta) \quad \forall \eta$$
MetaGrad Algorithm

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

\[
\ell^\eta_t(u) = \eta(u - w_t)^T g_t + \eta^2(u - w_t)^T g_t g_t^T (u - w_t)
\]

One **Slave** algorithm per \( \eta \) produces \( w^\eta_t \) such that

\[
\sum_{t=1}^T \ell^\eta_t(w^\eta_t) - \sum_{t=1}^T \ell^\eta_t(u) \leq R^u_{\text{slave}}(\eta)
\]

Single **Master** algorithm produces \( w_t \) such that

\[
\underbrace{\sum_{t=1}^T \ell^\eta_t(w_t) - \sum_{t=1}^T \ell^\eta_t(w^\eta_t)}_{=0} \leq R_{\text{master}}(\eta) \quad \forall \eta
\]

Together:

\[
- \sum_{t=1}^T \ell^\eta_t(u) \leq R^u_{\text{slave}}(\eta) + R_{\text{master}}(\eta) \quad \forall \eta
\]
MetaGrad Algorithm

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

$$
\ell^n_t(u) = \eta(u - w_t)^T g_t + \eta^2(u - w_t)^T g_t g_t^T (u - w_t)
$$

One **Slave** algorithm per $\eta$ produces $w^n_t$ such that

$$
\sum_{t=1}^{T} \ell^n_t(w^n_t) - \sum_{t=1}^{T} \ell^n_t(u) \leq R^u_{\text{slave}}(\eta)
$$

Single **Master** algorithm produces $w_t$ such that

$$
\underbrace{\sum_{t=1}^{T} \ell^n_t(w_t) - \sum_{t=1}^{T} \ell^n_t(w^n_t)}_{=0} \leq R^u_{\text{master}}(\eta) \quad \forall \eta
$$

Together: $- \sum_{t=1}^{T} \ell^n_t(u) \leq R^u_{\text{slave}}(\eta) + R^u_{\text{master}}(\eta) \quad \forall \eta$

$$
\sum_{t=1}^{T} (w_t - u)^T g_t \leq \frac{R^u_{\text{slave}}(\eta) + R^u_{\text{master}}(\eta)}{\eta} + \eta V^u_T
$$
MetaGrad Algorithm

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

$$
\ell^n_t(u) = \eta(u - w_t)^T g_t + \eta^2(u - w_t)^T g_t g_t^T(u - w_t)
$$

One **Slave** algorithm per $\eta$ produces $w_t^n$ such that

$$
\sum_{t=1}^{T} \ell^n_t(w_t^n) - \sum_{t=1}^{T} \ell^n_t(u) \leq R_{\text{slave}}^u(\eta)
$$

Single **Master** algorithm produces $w_t$ such that

$$
\sum_{t=1}^{T} \ell^n_t(w_t) - \sum_{t=1}^{T} \ell^n_t(w_t^n) \leq R_{\text{master}}(\eta) \quad \forall \eta
$$

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

$$
\sum_{t=1}^{T} (w_t - u)^T g_t \leq \frac{O(d \ln T) + O(\ln \ln T)}{\eta} + \eta V_T^u
$$
MetaGrad Algorithm

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

$$\ell^n_t(u) = \eta(u - w_t)^T g_t + \eta^2 (u - w_t)^T g_t g_t^T (u - w_t)$$

One **Slave** algorithm per $\eta$ produces $w^\eta_t$ such that

$$\sum_{t=1}^T \ell^n_t(w^\eta_t) - \sum_{t=1}^T \ell^n_t(u) \leq R_{\text{slave}}^u(\eta)$$

Single **Master** algorithm produces $w_t$ such that

$$\left\{ \begin{array}{c}
\sum_{t=1}^T \ell^n_t(w_t) - \sum_{t=1}^T \ell^n_t(w^\eta_t) \\ = 0
\end{array} \right\} \leq R_{\text{master}}(\eta) \quad \forall \eta$$

Together:

$$- \sum_{t=1}^T \ell^n_t(u) \leq R_{\text{slave}}^u(\eta) + R_{\text{master}}(\eta) \quad \forall \eta$$

$$\sum_{t=1}^T (w_t - u)^T g_t \leq \frac{O(d \ln T) + O(\ln \ln T)}{\eta} + \eta V_T^u \Rightarrow O\left(\sqrt{V_T^u d \ln T}\right)$$
MetaGrad Master

Goal: aggregate slave predictions \( \mathbf{w}_t^\eta \) for all \( \eta \) 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 \( \ell_t^\eta \) for all \( \eta \) simultaneously.

Compute **exponential weights** with performance of each \( \eta \) measured by its own surrogate loss:

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

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

\[
\mathbf{w}_t = \frac{\sum_\eta \pi_t(\eta) \eta \mathbf{w}_t^\eta}{\sum_\eta \pi_t(\eta) \eta}
\]
Potential

\[ \Phi_T = \sum_{\eta} \pi_1(\eta) e^{-\sum_{t=1}^{T} \ell_t^\eta(w_t^\eta)} \]

Proof outline:

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

\[ \pi_1(\eta) e^{-\sum_{t=1}^{T} \ell_t^\eta(w_t^\eta)} \leq 1 \quad \forall \eta \]

\[ \sum_{t=1}^{T} \ell_t^\eta(w_t) - \sum_{t=1}^{T} \ell_t^\eta(w_t^\eta) \leq -\ln \pi_1(\eta) \]

\[ = 0 \]
MetaGrad Master Analysis

Potential

\[ \Phi_T = \sum_{\eta} \pi_1(\eta) e^{-\sum_{t=1}^{T} \ell^n_t(w^n_t)} \]

Proof outline:

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

\[ \pi_1(\eta) e^{-\sum_{t=1}^{T} \ell^n_t(w^n_t)} \leq 1 \quad \forall \eta \]

\[ \sum_{t=1}^{T} \ell^n_t(w_t) - \sum_{t=1}^{T} \ell^n_t(w^n_t) \leq -\ln \pi_1(\eta) \]

\[ = 0 \]

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

\[ -\ln \pi_1(\eta) = O(\ln \ln T) \]
Surrogate loss $\ell_t^{\eta}(u) = \eta(u - w_t)^T g_t + \eta^2 (u - w_t)^T g_t g_t^T (u - w_t)$ is \textit{exp-concave}, even if $f_t$ is not.

Upper bound by tangent at $u = w_t$:

$$e^{-\ell_t^{\eta}(u)} \leq 1 + \eta (w_t - u)^T g_t$$
MetaGrad Master Analysis: Decreasing Potential

Surrogate loss $\ell^n_t(u) = \eta(u - w_t)^\top g_t + \eta^2(u - w_t)^\top g_t g_t^\top (u - w_t)$ is \textit{exp-concave}, even if $f_t$ is not.

Upper bound by tangent at $u = w_t$:

$$e^{-\ell^n_t(u)} \leq 1 + \eta (w_t - u)^\top g_t$$

Choose master’s weights to ensure decreasing potential:

$$\Phi_T - \Phi_{T-1} = \sum_\eta \pi_1(\eta) e^{-\sum_{t<T} \ell^n_t(w^n_t)} \left( e^{-\ell^n_T(w^n_T)} - 1 \right)$$

$$\leq \sum_\eta \pi_1(\eta) e^{-\sum_{t<T} \ell^n_t(w^n_t)} \eta (w_T - w^n_T)^\top g_T$$

$$= 0 \quad \text{for any } g_T$$
MetaGrad Slave

**Goal:** Given \( \eta \), minimize regret w.r.t. exp-concave surrogate \( \ell_t^\eta \).

**Update:**

\[
\tilde{w}_{t+1}^\eta = w_t^\eta - \eta s_t^\eta \sum_{t+1}^\eta g_t,
\]

where

\[
\sum_{t+1}^\eta = \left( \frac{1}{D^2} I + 2\eta^2 \sum_{s=1}^{t} g_s g_s^\top \right)^{-1} \quad s_t^\eta = 1 + 2\eta g_t^\top (w_t^\eta - w_t)
\]

**Project onto domain:**

\[
w_{t+1}^\eta = \arg \min_{u \in \mathcal{U}} (u - \tilde{w}_{t+1}^\eta)^\top (\sum_{t+1}^\eta)^{-1} (u - \tilde{w}_{t+1}^\eta)
\]

**If master = slave, i.e.**

\( w_{t+1}^\eta = w_t \), **then is**

**Online Newton Step**
MetaGrad Slave

**Goal:** Given $\eta$, minimize regret w.r.t. exp-concave surrogate $\mathcal{L}_t^\eta$.

**Update:**

$$\tilde{w}_{t+1}^\eta = w_t^\eta - \eta s_t^\eta \Sigma_{t+1}^\eta g_t,$$

where

$$\Sigma_{t+1}^\eta = \left( \frac{1}{D^2} I + 2\eta^2 \sum_{s=1}^t g_s g_s^\top \right)^{-1}$$

$$s_t^\eta = 1 + 2\eta g_t^\top (w_t^\eta - w_t)$$

**Project onto domain:**

$$w_{t+1}^\eta = \arg\min_{u \in U} (u - \tilde{w}_{t+1}^\eta)^\top (\Sigma_{t+1}^\eta)^{-1} (u - \tilde{w}_{t+1}^\eta)$$

- If master = slave, i.e. $w_t^\eta = w_t$, then is **Online Newton Step**
Exponential Weights

- Continuous set of experts $u \in \mathbb{R}^d$
- Gaussian prior $P_1 = \mathcal{N}(0, D^2 I)$

\[
\text{d} \tilde{P}_{t+1}(u) = \frac{e^{-\ell_t^n(u)}dP_t(u)}{Z} \quad \text{(update)}
\]

\[
P_{t+1} = \min_{P: \mu_P \in \mathcal{U}} \text{KL}(P \| \tilde{P}_{t+1}) \quad \text{(project)}
\]

Play the mean of exponential weights:

\[
\tilde{P}_t = \mathcal{N}(\tilde{w}_t^n, \Sigma_t^n)
\]

\[
P_t = \mathcal{N}(w_t^n, \Sigma_t^n)
\]

- Can understand **Online Newton Step** and many other algorithms this way
MetaGrad Slave Analysis

Standard exponential weights analysis gives regret bound in space of distributions for all $Q$:

$$
\text{KL}(Q\|P_1) \geq \sum_{t=1}^{T} - \ln \mathbb{E}_{P_t}[e^{-\eta_t(u)}] - \sum_{t=1}^{T} \mathbb{E}_{Q}[\ell^n_t(u)]
$$

$$
\text{exp-conc} \geq \sum_{t=1}^{T} \ell^n_t(\mu_{P_t}) - \sum_{t=1}^{T} \mathbb{E}_{Q}[\ell^n_t(u)]
$$
MetaGrad Slave Analysis

Standard exponential weights analysis gives regret bound in space of distributions for all $Q$:

$$\text{KL}(Q\|P_1) \geq \sum_{t=1}^{T} - \ln \mathbb{E}_{P_t}[e^{-\ell_t^\eta(u)}] - \sum_{t=1}^{T} \mathbb{E}_{Q}[\ell_t^\eta(u)]$$

with $\text{exp-conc}$:

$$\geq \sum_{t=1}^{T} \ell_t^\eta(\mu_{P_t}) - \sum_{t=1}^{T} \mathbb{E}_{Q}[\ell_t^\eta(u)]$$

Specialize to $Q = \mathcal{N}(u^*, D^2\Sigma) + \text{algebra}$:

$$\frac{1}{2D^2} \|u^*\|^2 + \frac{1}{2}(-\ln \det(\Sigma) + \text{tr}(\Sigma) - d)$$

$$\geq \sum_{t=1}^{T} \ell_t^\eta(\mu_{P_t}) - \sum_{t=1}^{T} \ell_t^\eta(u^*) - \sum_{t=1}^{T} \eta^2 D^2 \text{tr}(\Sigma g_t g_t^\top)$$
MetaGrad Slave Analysis

Standard exponential weights analysis gives regret bound in space of distributions for all $Q$:

$$\text{KL}(Q\|P_1) \geq \sum_{t=1}^{T} - \ln \mathbb{E}_{P_t}[e^{-\ell_t^n(u)}] - \sum_{t=1}^{T} \mathbb{E}_Q[\ell_t^n(u)]$$

$$\geq \sum_{t=1}^{T} \ell_t^n(\mu_{P_t}) - \sum_{t=1}^{T} \mathbb{E}_Q[\ell_t^n(u)]$$

Specialize to $Q = \mathcal{N}(u^*, D^2\Sigma) + \text{algebra}$:

$$\frac{1}{2D^2} \|u^*\|^2 + \frac{1}{2} \left( - \ln \det(\Sigma) + \text{tr}(\Sigma) - d \right)$$

$$\geq \sum_{t=1}^{T} \ell_t^n(\mu_{P_t}) - \sum_{t=1}^{T} \ell_t^n(u^*) - \sum_{t=1}^{T} \eta^2 D^2 \text{tr}(\Sigma g_t g_t^\top)$$

Optimize $\Sigma$:

$$R_{\text{slave}}^u(\eta) \leq \frac{1}{2D^2} \|u^*\| + \frac{1}{2} \ln \det \left( I + 2\eta^2 D^2 \sum_{t=1}^{T} g_t g_t^\top \right) = O(d \ln T)$$
Outline

Two General Fast Rate Conditions

MetaGrad Algorithm

Exponential Weights Interpretation of Online Learning Algorithms
Many Algorithms as Exponential Weights

Exponentiated Gradient [Kivinen and Warmuth, 1997]

- Continuous set of experts $u \in \mathbb{R}^d$ with loss $\ell_t(u) = \langle u, g_t \rangle$
- Prior $P_1$ puts point masses on corners of probability simplex

$$dP_{t+1}(u) = \frac{e^{-\eta \ell_t(u)}dP_t(u)}{Z} \implies \mathbb{E}_{P_{t+1}}[u] = P_{t+1} = \text{EG}$$
Many Algorithms as Exponential Weights

Exponentiated Gradient [Kivinen and Warmuth, 1997]

- Continuous set of experts \( u \in \mathbb{R}^d \) with loss \( \ell_t(u) = \langle u, g_t \rangle \)
- Prior \( P_1 \) puts point masses on corners of probability simplex

\[
dP_{t+1}(u) = \frac{e^{-\eta \ell_t(u)}dP_t(u)}{Z} \implies \mathbb{E}_{P_{t+1}}[u] = P_{t+1} = \text{EG}
\]

Gradient Descent

Gaussian prior \( P_1 = \mathcal{N}(0, I) \) \( \implies \) \( P_{t+1}(u) = \mathcal{N}(-\eta \sum_{t=1}^{T} g_t, I) \)
Many Algorithms as Exponential Weights

Exponentiated Gradient [Kivinen and Warmuth, 1997]

- Continuous set of experts \( u \in \mathbb{R}^d \) with loss \( \ell_t(u) = \langle u, g_t \rangle \)
- Prior \( P_1 \) puts point masses on corners of probability simplex

\[
dP_{t+1}(u) = \frac{e^{-\eta \ell_t(u)}dP_t(u)}{Z} \quad \Rightarrow \quad \mathbb{E}_{P_{t+1}}[u] = P_{t+1} = \text{EG}
\]

Gradient Descent

Gaussian prior \( P_1 = \mathcal{N}(0, I) \) \( \Rightarrow \) \( P_{t+1}(u) = \mathcal{N}(-\eta \sum_{t=1}^{T} g_t, I) \)

Mirror Descent [Van der Hoeven, 2016]

- Generalize to arbitrary prior \( P_1 \)
- \( \Phi(\theta) = \ln \int e^{\langle \theta, u \rangle} dP_1(u) \):
  - CGF for exponential family with carrier \( P_1 \)

\[
\mu_{P_{t+1}} = \nabla \Phi \left( \nabla \Phi^* (\mu_{P_t}) - \eta g_t \right)
\]

(update in natural parameters)
Summary and Last Remarks

MetaGrad

- $\tilde{O}(\sqrt{T})$ regret
- **Fast rates** (often $O(d \ln T)$) for:
  - Adversarial: exp-concave, strongly convex, fixed functions
  - Stochastic: under Bernstein condition (including for hinge loss)
Summary and Last Remarks

MetaGrad

- $\tilde{O}(\sqrt{T})$ regret
- **Fast rates** (often $O(d \ln T)$) for:
  - Adversarial: exp-concave, strongly convex, fixed functions
  - Stochastic: under Bernstein condition (including for hinge loss)

MetaGrad Algorithm

- **Master:**
  - Pays only $O(\ln \ln T)$ for learning the $\eta$ that is empirically optimal on the data
  - Almost exponential weights for surrogate loss, but need to **tilt towards larger learning rates**

- **Slave:**
  - Continuous exponential weights on surrogate loss
  - Matrix updates take $O(d^2)$ work, projections often $O(d^3)$
  - Open problem: add sketching like [Luo et al., 2016]?
Summary and Last Remarks

MetaGrad

- $\tilde{O}(\sqrt{T})$ regret
- **Fast rates** (often $O(d \ln T)$) for:
  - Adversarial: exp-concave, strongly convex, fixed functions
  - Stochastic: under Bernstein condition (including for hinge loss)

MetaGrad Algorithm

- Master:
  - Pays only $O(\ln \ln T)$ for learning the $\eta$ that is empirically optimal on the data
  - Almost exponential weights for surrogate loss, but need to **tilt towards larger learning rates**

- Slave:
  - Continuous exponential weights on surrogate loss
  - Matrix updates take $O(d^2)$ work, projections often $O(d^3)$
  - Open problem: add sketching like [Luo et al., 2016]?

Cool Aside: View many online algorithms as continuous exponential weights
References


