Learning the Learning Rate for Prediction with Expert Advice

Wouter M. Koolen  Tim van Erven  Peter D. Grünwald

Lorentz Workshop Leiden, Thursday 20th November, 2014
Online Learning Algorithms

- Work in Practice
- Theoretical Performance Guarantees

?
Learning as a Game

- worst-case safe algorithm

Regret vs. problem instances:
- High (bad)
- Minimax
- 0 (perfect)
Practice is not Adversarial

- worst-case safe algorithm
- special-purpose algorithm

regret

problem instances

minimax

high (bad)

0 (perfect)
Luckiness

[Graph showing regret vs. problem instances with labels for worst-case safe algorithm, special-purpose algorithm, high (bad), minimax, and 0 (perfect)]
Fundamental model for learning: Hedge setting

- $K$ experts

$L$ learner plays distribution $w_t = (w_1^t, ..., w_K^t)$ on experts

Adversary reveals expert losses $\ell_t = (\ell_1^t, ..., \ell_K^t) \in [0, 1]^K$

Learner incurs loss $w_t^\top \ell_t$

Evaluation criterion is the regret:

$$R_T := T \sum_{t=1}^T w_t^\top \ell_t - \min_k T \sum_{t=1}^T \ell_k^t$$
Fundamental model for learning: Hedge setting

- **$K$ experts**

- In round $t = 1, 2, \ldots$
  - Learner plays distribution $\mathbf{w}_t = (w_1^t, \ldots, w_K^t)$ on experts
  - Adversary reveals expert losses $\ell_t = (\ell_1^t, \ldots, \ell_K^t) \in [0, 1]^K$

- Learner incurs loss $\mathbf{w}_t^T \ell_t$
Fundamental model for learning: Hedge setting

- \( K \) experts

- In round \( t = 1, 2, \ldots \)
  - Learner plays distribution \( \mathbf{w}_t = (w_1^t, \ldots, w_K^t) \) on experts
  - Adversary reveals expert losses \( \ell_t = (\ell_1^t, \ldots, \ell_K^t) \in [0, 1]^K \)

- Learner incurs loss \( \mathbf{w}_t^T \ell_t \)

- Evaluation criterion is the regret:

\[
\mathcal{R}_T := \sum_{t=1}^{T} \mathbf{w}_t^T \ell_t - \min_k \sum_{t=1}^{T} \ell_k^t
\]

\( \underbrace{\text{Learner}}_{T} \) - \( \underbrace{\text{best expert}}_{T} \)
Canonical algorithm for the Hedge setting

Hedge algorithm with learning rate $\eta$:

$$w_t^k := \frac{e^{-\eta L_{t-1}^k}}{\sum_k e^{-\eta L_{t-1}^k}}$$

where

$$L_{t-1}^k = \sum_{s=1}^{t-1} \ell_s^k.$$
Canonical algorithm for the Hedge setting

Hedge algorithm with learning rate $\eta$:

$$w^k_t := \frac{e^{-\eta L^k_{t-1}}}{\sum_k e^{-\eta L^k_{t-1}}} \quad \text{where} \quad L^k_{t-1} = \sum_{s=1}^{t-1} \ell^k_s.$$ 

The tuning $\eta = \eta_{\text{worst case}} := \sqrt{\frac{8 \ln K}{T}}$ results in

$$R_T \leq \sqrt{T/2 \ln K}$$

and we have matching lower bounds.
Canonical algorithm for the Hedge setting

Hedge algorithm \textbf{with learning rate } $\eta$:

\[
w^k_t := \frac{e^{-\eta L^k_{t-1}}}{\sum_k e^{-\eta L^k_{t-1}}} \quad \text{where} \quad L^k_{t-1} = \sum_{s=1}^{t-1} \ell^k_s.
\]

The tuning $\eta = \eta^{\text{worst case}} := \sqrt{\frac{8 \ln K}{T}}$ results in

\[
\mathcal{R}_T \leq \sqrt{\frac{T}{2 \ln K}}
\]

and we have matching lower bounds.

Case closed?
Practitioners report that tuning $\eta \gg \eta_{\text{worst case}}$ works much better. [DGGS13]
Theory and Practice getting closer

Practitioners report that tuning $\eta \gg \eta_{\text{worst case}}$ works much better. [DGGS13]

Series of worst-case **data-dependent** improvements

$$R_T \leq \sqrt{T/2 \ln K}$$
Theory and Practice getting closer

Practitioners report that tuning $\eta \gg \eta_{\text{worst case}}$ works much better. [DGGS13]

Series of worst-case data-dependent improvements

$$\mathcal{R}_T \leq \sqrt{\frac{T}{2 \ln K}}$$

and extension to scenarios where Follow-the-Leader ($\eta = \infty$) shines (IID losses)

$$\mathcal{R}_T \leq \min \{ \mathcal{R}_T^{\text{worst case}}, \mathcal{R}_T^{\infty} \}$$
Practitioners report that tuning $\eta \gg \eta_{\text{worst case}}$ works much better. [DGGS13]

Series of worst-case data-dependent improvements

$$R_T \leq \sqrt{T/2 \ln K}$$

and extension to scenarios where Follow-the-Leader ($\eta = \infty$) shines (IID losses)

$$R_T \leq \min\{R_{T \text{ case}}, R_{T \infty}\}$$

Case closed?
Grand goal: be almost as good as best learning rate $\eta$

\[ R_T \approx \min_{\eta} R_{T}^{\eta}. \]

- Example problematic data
- Key ideas
Current $\eta$ tunings miss the boat

$T = 100000$
Current $\eta$ tunings miss the boat

$T = 100000$

$R_{\eta}^T$

Bad expert

<table>
<thead>
<tr>
<th>expert</th>
<th>rounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:</td>
<td>1 1 1 1 1 1 ...</td>
</tr>
</tbody>
</table>
Current $\eta$ tunings miss the boat

$T = 100000$

- Bad expert
- FTL worst case

<table>
<thead>
<tr>
<th>expert</th>
<th>rounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>0 1 0 1 0 1 0 ...</td>
</tr>
<tr>
<td>2:</td>
<td>1 0 1 0 1 0 1 ...</td>
</tr>
</tbody>
</table>

$\mathcal{R}_T$ vs $\eta$ graph with different strategies and their corresponding performance metrics.
Current $\eta$ tunings miss the boat

$T = 100000$

![Graph showing different expert performances with $\eta$ tuning](image)
Current $\eta$ tunings miss the boat

$T = 100000$

$R_\eta^n$

- Bad expert
- FTL worst case
- WC-eta killer
- Combined

$\eta$
LLR algorithm in a nutshell

LLR

- maintains a finite grid $\eta^1, \ldots, \eta^{i_{\text{max}}}, \eta^{\text{ah}}$
- cycles over the grid. For each $\eta^i$:
  - Play the $\eta^i$ Hedge weights
  - Evaluate $\eta^i$ by its mixability gap
  - Until its budget doubled
- adds next lower grid point on demand

Resources:

- Time: $O(K)$ per round (same as Hedge).
- Memory: $O(\ln T) \rightarrow O(1)$. 
LLR algorithm in a nutshell

LLR

- maintains a finite grid $\eta^1, \ldots, \eta^{i_{\text{max}}}, \eta^{\text{ah}}$
- cycles over the grid. For each $\eta^i$:
  - Play the $\eta^i$ Hedge weights
  - Evaluate $\eta^i$ by its mixability gap
  - Until its budget doubled
- adds next lower grid point on demand

Resources:
- Time: $O(K)$ per round (same as Hedge).
- Memory: $O(\ln T) \rightarrow O(1)$. 
Unavoidable notation

\[ h_t = w_t^T \ell_t, \]  

(Hedge loss)

\[ m_t = \frac{-1}{\eta} \ln \sum_k w_k^t e^{-\eta \ell_t^k}, \]  

(Mix loss)

\[ \delta_t = h_t - m_t. \]  

(Mixability gap)
Unavoidable notation

\[ h_t = w_t^T \ell_t, \quad \text{ (Hedge loss)} \]

\[ m_t = -\frac{1}{\eta} \ln \sum_k w_t^k e^{-\eta \ell_t^k}, \quad \text{ (Mix loss)} \]

\[ \delta_t = h_t - m_t. \quad \text{ (Mixability gap)} \]

And capitals denote cumulatives

\[ \Delta_T = \sum_{t=1}^{T} \delta_t, \ldots \]
Key Idea 1: Monotone regret lower bound

Problem: Regret $R^n_T$ is not increasing with $T$.

But we have a monotone lower bound:

$$R^n_T \geq \Delta^n_T$$

Proof:

$$R^n_t = H_T - L^*_T = H_T - M_T + M_T - L^*_T$$

Now use

$$M_T = -\frac{1}{\eta} \ln \left( \sum_K \frac{1}{K} e^{-\eta L^*_T} \right) \in L^*_T + \left[ 0, \frac{\ln K}{\eta} \right]$$

Upshot: measure quality of each $\eta$ using cumulative mixability gap.
Key Idea 2: Grid of $\eta$ suffices

For $\gamma \geq 1$:

$$\delta_t^{\gamma\eta} \leq \gamma e^{(\gamma-1)(\ln K + \eta)} \delta_t^\eta$$

I.e. $\delta_t^\eta$ cannot be much better than $\delta_t^{\gamma\eta}$.

Exponentially spaced grid of $\eta$ suffices.
Key Idea 3: Lowest $\eta$ is “AdaHedge”

AdaHedge:

$$\eta_{ah}^t := \frac{\ln K}{\Delta_{t-1}^{ah}}$$

Result:

$$\mathcal{R}_T \leq \sum_{i=1}^{i_{max}} \Delta_i^T + c\Delta_T^{ah}$$
Key Idea 4: Budgeted timesharing

Active grid points
\[ \eta^1, \eta^2, \ldots, \eta^{i_{\text{max}}}, \eta^{\text{ah}} \]

with (heavy-tailed) prior distribution
\[ \pi^1, \pi^2, \ldots, \pi^{i_{\text{max}}}, \pi^{\text{ah}} \]

LLR maintains \textbf{invariant}:
\[ \frac{\Delta^1_T}{\pi^1} \approx \frac{\Delta^2_T}{\pi^2} \approx \ldots \approx \frac{\Delta^{i_{\text{max}}}_T}{\pi^{i_{\text{max}}}} \approx \frac{\Delta^{\text{ah}}_T}{\pi^{\text{ah}}} \]

Run each \( \eta_i \) in turn until its cumulative mixability gap \( \frac{\Delta^i_T}{\pi^i} \) doubled.
\[ \sum_{i=1}^{i_{\text{max}}} \Delta^i_T = \sum_{i=1}^{i_{\text{max}}} \pi^i \frac{\Delta^i_T}{\pi^i} \approx \frac{\Delta^j_T}{\pi^j} \sum_{i=1}^{i_{\text{max}}} \pi^i \leq \frac{\Delta^j_T}{\pi^j} \]
Putting it all together

Two bounds:

\[ \mathcal{R}_T \leq \tilde{O} \left\{ \ln K \ln \frac{1}{\eta} \mathcal{R}_T^\eta \quad \text{for all } \eta \in [\eta_{t^*}^{ah}, 1] \right\} \]
Run on synthetic data \( (T = 2 \cdot 10^7) \)

- Worst-case bound and worst-case \( \eta \)
- Hedge\(^{(\eta)}\)
- AdaHedge
- FlipFlop
- LLR and \( \eta_{t^*}^{ah} \)
Conclusion

- Higher learning rates often achieve lower regret
  - In practice
  - Constructed data
- Learning the Learning Rate (LLR) algorithm
  - Performance close to best learning rate in hindsight
Conclusion

- Higher learning rates often achieve lower regret
  - In practice
  - Constructed data
- Learning the Learning Rate (LLR) algorithm
  - Performance close to best learning rate in hindsight

Open problems:
- LLR as PoC
  Can we do it simpler, prettier, smoother and tighter?
Thank you!
Marie Devaine, Pierre Gaillard, Yannig Goude, and Gilles Stoltz.