Learning the Learning Rate for Prediction with Expert Advice

Wouter M. Koolen Tim van Erven Peter D. Grünwald

Queensland University of Technology · CWI Centrum Wiskunde & Informatica · Universiteit Leiden · Berkeley University of California

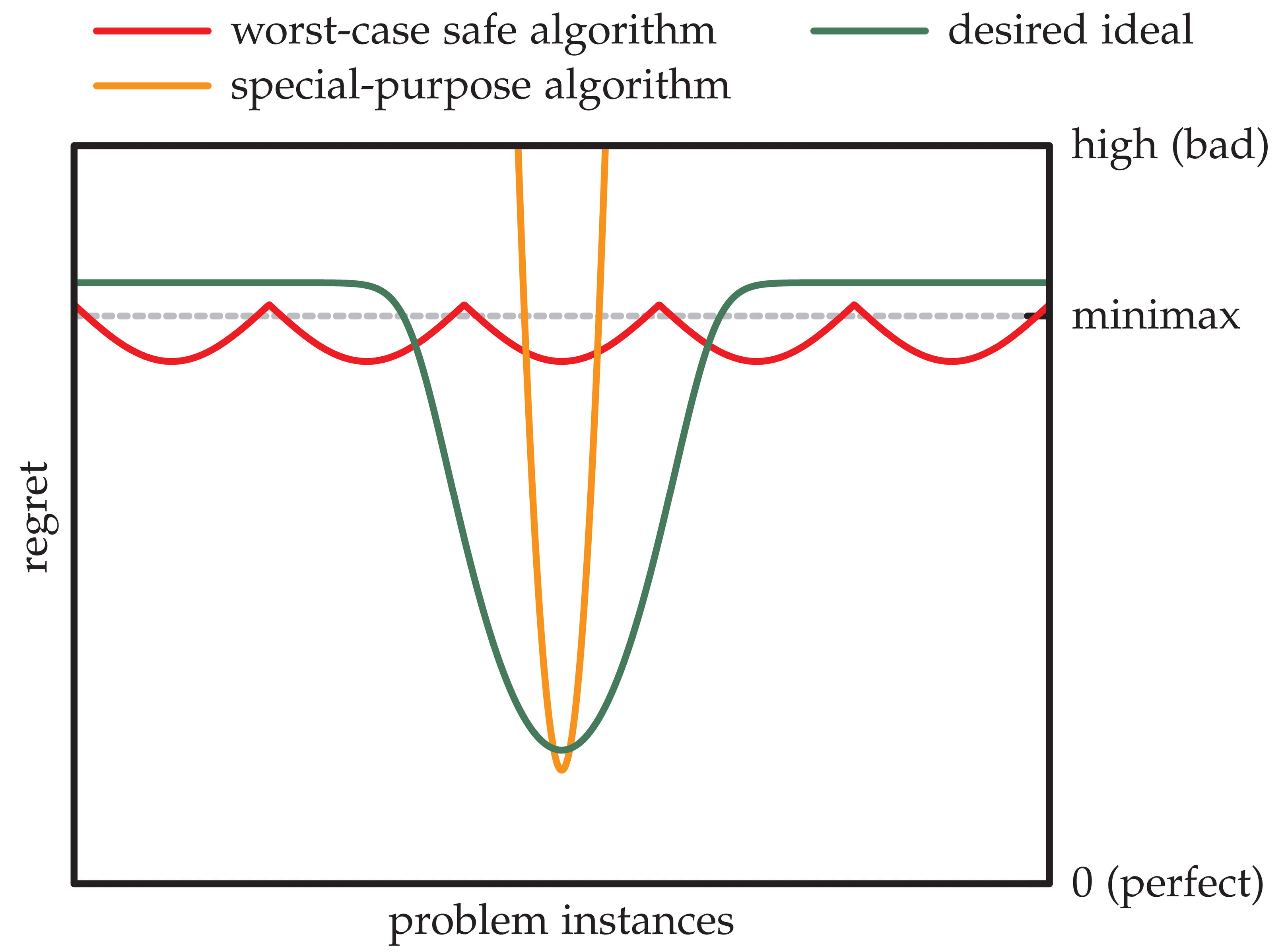
NIPS Mon20 December 8th 2014

Summary

In online learning, tuning the learning rate is **essential** but **tricky**.
We present LLR, a new algorithm for setting the learning rate that

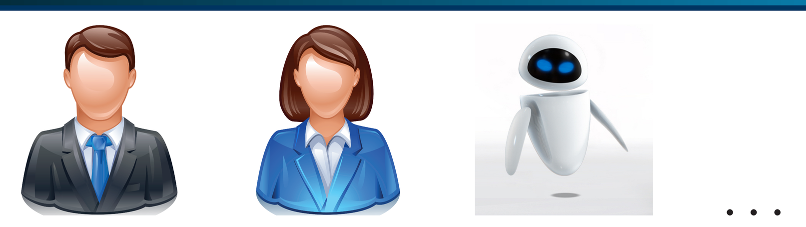
- performs as well as the **best learning rate in hindsight**, and
- is **as fast as** the standard exponential weights algorithm.

The problem



Fundamental model for learning: Hedge setting

K experts



In round $t = 1, 2, \dots$

- Learner plays distribution $\boldsymbol{w}_t = (w_t^1, \dots, w_t^K)$ on experts
- Adversary reveals expert losses $\boldsymbol{\ell}_t = (\ell_t^1, \dots, \ell_t^K) \in [0,1]^K$

- Learner incurs loss $\boldsymbol{w}_t^\intercal \boldsymbol{\ell}_t$

Evaluation criterion is the **regret** w.r.t. the best expert in hindsight:

$$\mathcal{R}_T \;:=\; \underbrace{\sum_{t=1}^T \boldsymbol{w}_t^\intercal \boldsymbol{\ell}_t}_{\text{Learner}} - \underbrace{\min_k \sum_{t=1}^T \ell_t^k}_{\text{best expert}}$$

The exponential weights algorithm

The Hedge or exponential weights algorithm **with learning rate η**:

$$w_t^k \;:=\; \frac{e^{-\eta L_{t-1}^k}}{\sum_k e^{-\eta L_{t-1}^k}} \qquad \text{where} \qquad L_{t-1}^k \;:=\; \sum_{s=1}^{t-1} \ell_s^k.$$

The worst-case safe tuning $\eta = \sqrt{\frac{8 \ln K}{T}}$ results in

$$\mathcal{R}_T \;\leq\; \sqrt{T/2 \ln K}.$$

tight for worst-case data

Prior work on adaptive tuning

Second-order Bounds [Cesa-Bianchi, Mansour, Stoltz ML'07]:

$$\mathcal{R}_T \;\leq\; O\left(\sqrt{V_T \ln K}\right), \tag{1}$$

where $V_T = \sum_{t=1}^T v_t$ and v_t is variance of loss ℓ_t^k when $k \sim \boldsymbol{w}_t$.

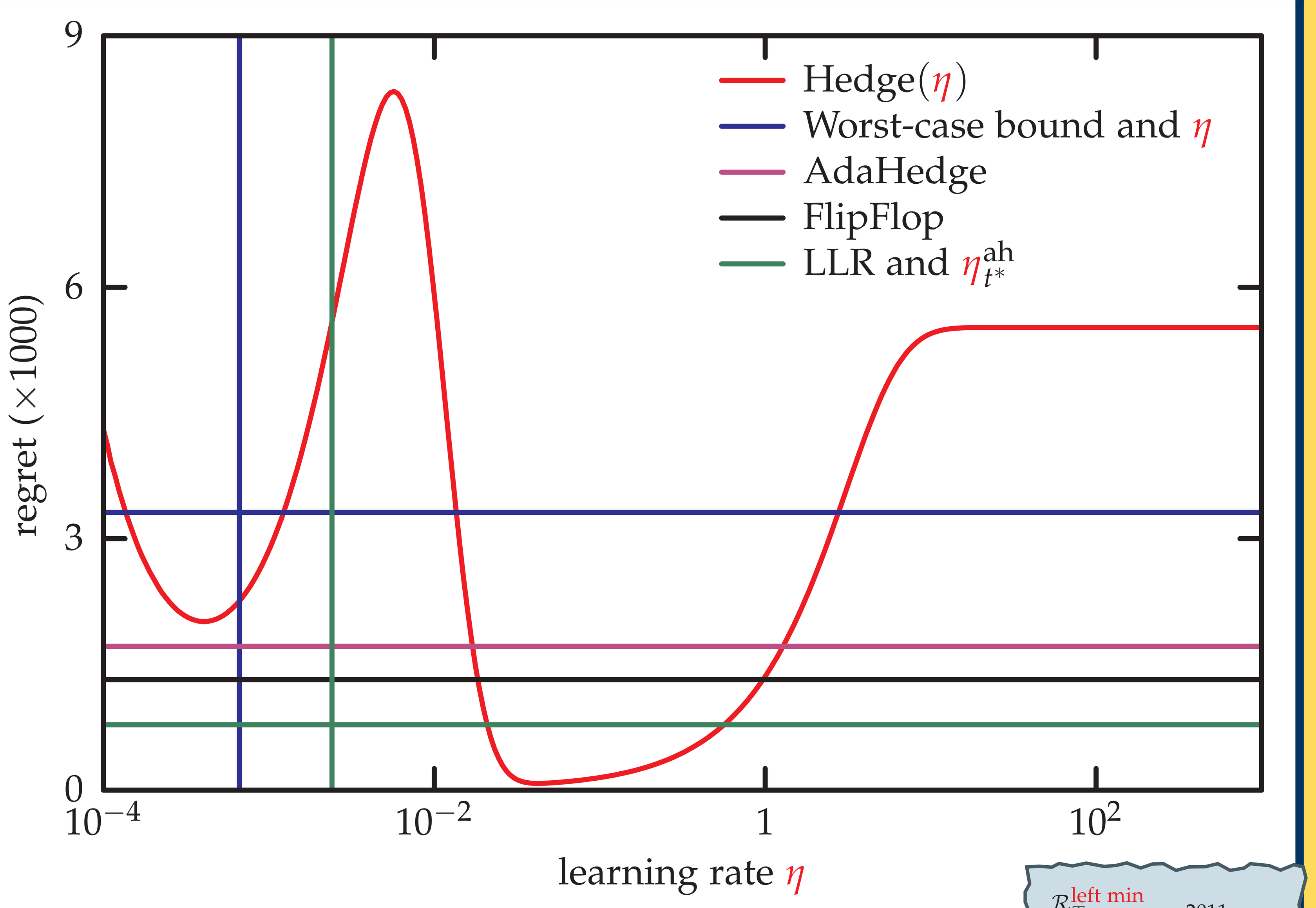
FlipFlop [De Rooij and us, JMLR'14]:
Satisfies (1) and is never worse than Follow-the-Leader ($\eta = \infty$):

$$\mathcal{R}_T \;\leq\; C \mathcal{R}_T^\infty. \tag{2}$$

Big question

Can we compete with all η?

Intermediate learning rates can be crucial



$$\frac{\mathcal{R}_T^{\text{left min}}}{\mathcal{R}_T^{\text{best}}} \approx \frac{2011}{81} \approx 25$$

New results

The Learning the Learning Rate (LLR) algorithm satisfies (1) and (2) and in addition

$$\mathcal{R}_T \;\leq\; \tilde{O}\left(\ln K \ln \tfrac{1}{\eta}\right) \mathcal{R}_T^\eta \qquad \text{for all } \eta \in [\eta_{t^*}^{\text{ah}}, 1].$$

- Trumps $\sqrt{T}$ additive overhead.
- In above range $\ln \frac{1}{\eta} = O(\ln T)$.
- So $\mathcal{R}_T = \tilde{O}(T^\alpha)$ whenever $\min_\eta \mathcal{R}_T^\eta = O(T^\alpha)$ for some $\alpha > 0$.

LLR algorithm in a nutshell

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

η_t **not monotonic** in t

Resource consumption: same as Hedge (trick!)

LLR algorithm key ideas

- Basic idea: budgeted timesharing. Cycle through ηs from an exponentially spaced grid to keep their regrets approximately equal. Then our regret is
$$\mathcal{R}_T \;\leq\; \#\text{gridpoints} \times \mathcal{R}_T^{\text{best } \eta}.$$
- Refinement: Use prior distribution on ηs to get
$$\mathcal{R}_T \;\leq\; \frac{\mathcal{R}_T^{\text{best } \eta}}{\text{prior of best } \eta}$$
independent of the number of grid points.
- Problem: does not quite work, because regret $\mathcal{R}_T^\eta$ for fixed η is not monotonically increasing with time T
- Solution: cumulative mixability gap $\Delta_T^\eta = \sum_{t=1}^T \delta_t^\eta$ where
$$\delta_t^\eta \;=\; \text{Hedge loss} - \text{mix loss} \;=\; \boldsymbol{w}_t^\intercal \boldsymbol{\ell}_t - \frac{-1}{\eta} \ln \sum_k w_t^k e^{-\eta \ell_t^k}$$
is a good proxy (tightest monotic lower bound on $\mathcal{R}_T^\eta$).
- Building block of independent interest:
$$\delta_t^{2\eta} \;\leq\; 6K\delta_t^\eta.$$

exponentially spacing η suffices