Learning Faster from Easy Data

Wouter M. Koolen

CWI Scientific Meeting, Friday 27th November, 2015
Bio

- 06–11 PhD at CWI (Centrum Wiskunde & Informatica)
- 11–13 Postdoc at Royal Holloway University of London
- 13–15 Postdoc at QUT Queensland University of Technology and Berkeley University of California
- 15– VENI at CWI (Centrum Wiskunde & Informatica)
Me  →  Learning  →  Information

A&C
This talk: online learning

Sequential decision making protocol

Definition

To learn $X = \text{act}$ as if you already know best $x \in X$
Typical online learning applications

- Invest like best stock (or portfolio)
- Predict demand like best linear regressor (Amazon)
- Commute like best route (OSP)
- Compress like best variable-order markov model (CTW)
- Tracking the best electricity consumption forecasting company (EDF)

...
Applications outside online learning comfort-zone

- Convex optimisation, both online, and batch (SGD).
- Computing Nash equilibria in two-player zero-sum games
- Game play (Monte Carlo Tree Search, e.g. for Go)
- Boosting
- Differential Privacy
- A/B testing
- Predictive complexity (algorithmic information theory)
- . . .
Fundamental model for learning: Hedge setting

- $K$ experts

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

Learner plays distribution $w^t = (w_1^t, \ldots, w_K^t)$ on experts

Learner observes expert losses $\ell^t = (\ell_1^t, \ldots, \ell_K^t) \in [0, 1]^K$

Learner incurs loss $w^\top \ell^t$

The goal is to have small regret $R_k^T := T \sum_{t=1}^{T} w^\top \ell^t - T \sum_{t=1}^{T} \ell_k^t$ with respect to every expert $k$.
Fundamental model for learning: Hedge setting

- $K$ experts

- In round $t = 1, 2, \ldots$
  - Learner plays distribution $w_t = (w_t^1, \ldots, w_t^K)$ on experts
  - Learner observes expert losses $\ell_t = (\ell_t^1, \ldots, \ell_t^K) \in [0, 1]^K$

- Learner incurs loss $w_t^T \ell_t$
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
  - Learner observes expert losses $\mathbf{\ell}_t = (\ell^1_t, \ldots, \ell^K_t) \in [0, 1]^K$

- Learner incurs loss $\mathbf{w}_t^T \mathbf{\ell}_t$

- The goal is to have small regret

$$R^k_T := \sum_{t=1}^{T} \mathbf{w}_t^T \mathbf{\ell}_t - \sum_{t=1}^{T} \ell^k_t$$

with respect to every expert $k$. 
Classic Hedge Result

The **Hedge** algorithm with **learning rate** $\eta$

$$w^{k}_{t+1} := \frac{e^{-\eta L^k_{t}}}{\sum_k e^{-\eta L^k_{t}}} \quad \text{where} \quad L^k_{t} = \sum_{s=1}^{t} \ell^k_{s},$$

upon proper tuning of $\eta$ ensures [Freund and Schapire, 1997] $R^k_{T} \prec \sqrt{T \ln K}$ for each expert $k$

which is tight for adversarial (worst-case) losses.
Classic Hedge Result

The **Hedge** algorithm with **learning rate** $\eta$

$$w_{t+1}^k := \frac{e^{-\eta L_t^k}}{\sum_k e^{-\eta L_t^k}} \quad \text{where} \quad L_t^k = \sum_{s=1}^t \ell_{ts}^k,$$

upon proper tuning of $\eta$ ensures [Freund and Schapire, 1997]

$$R_T^k \prec \sqrt{T \ln K}$$

for each expert $k$

which is tight for adversarial (worst-case) losses

but **underwhelming** in practice
The **Hedge** algorithm with **learning rate** $\eta$

\[
 w_{t+1}^k := \frac{e^{-\eta L_t^k}}{\sum_k e^{-\eta L_t^k}} \quad \text{where} \quad L_t^k = \sum_{s=1}^t \ell_{s}^{k},
\]

upon proper tuning of $\eta$ ensures [Freund and Schapire, 1997]

\[
 R_T^k \preceq \sqrt{T \ln K} \quad \text{for each expert} \ k
\]

which is tight for adversarial (worst-case) losses

but **underwhelming** in practice

▶ Why?
▶ Practitioners report good performance with ad-hoc $\eta$
▶ Can we do better?
Beyond the Worst Case

Two reasons data is often easier in practice:
Beyond the Worst Case

Two reasons data is often easier in practice:

<table>
<thead>
<tr>
<th>Data complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Stochastic data (gap)</td>
</tr>
<tr>
<td>▶ Low noise</td>
</tr>
<tr>
<td>▶ Low variance</td>
</tr>
</tbody>
</table>

Second-order & Quantiles
▶ Any combination
Beyond the Worst Case

Two reasons data is often **easier** in practice:

<table>
<thead>
<tr>
<th>Data complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Stochastic data (gap)</td>
</tr>
<tr>
<td>▶ Low noise</td>
</tr>
<tr>
<td>▶ Low variance</td>
</tr>
</tbody>
</table>

second-order
Beyond the Worst Case

Two reasons data is often easier in practice:

<table>
<thead>
<tr>
<th>Data complexity</th>
<th>Model complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Stochastic data (gap)</td>
<td>▶ Simple model is good</td>
</tr>
<tr>
<td>▶ Low noise</td>
<td>▶ Multiple good models</td>
</tr>
<tr>
<td>▶ Low variance</td>
<td></td>
</tr>
</tbody>
</table>

second-order
Beyond the Worst Case

Two reasons data is often easier in practice:

**Data complexity**
- Stochastic data (gap)
- Low noise
- Low variance

**Model complexity**
- Simple model is good
- Multiple good models

second-order

quantiles
Beyond the Worst Case

Two reasons data is often easier in practice:

**Data complexity**
- Stochastic data (gap)
- Low noise
- Low variance

**Model complexity**
- Simple model is good
- Multiple good models

Second-order & Quantiles
- Any combination
All we need is the right learning rate

Existing algorithms (Hedge, Prod, ...) with oracle learning rate $\eta$ exploit Sec-ord. & Quant.
All we need is the right learning rate

Existing algorithms (Hedge, Prod, ...) with oracle learning rate $\eta$ exploit Sec-ord. & Quant.

Can we exploit Second-order & Quantiles on-line?
But everyone struggles with the learning rate

Oracle $\eta$

* not monotonic,
* not smooth over time.


or Quantiles Hutter and Poland 2005, Chaudhuri, Freund, and Hsu 2009, Chernov and Vovk 2010, Luo and Schapire 2014
But everyone struggles with the learning rate

Oracle $\eta$

- **not** monotonic,
- **not** smooth

dergree over time.

State of the art:

**Second-order**


**Quantiles**

Learning the learning rate

With Tim van Erven: New framework for algorithm design where simply putting a prior $\gamma$ on $\eta$ and integrating it out works.

Our algorithm Squint

$$w^{k}_{t+1} \propto \pi(k) \mathbb{E}_{\gamma(\eta)} \left[ e^{\eta R^{k}_{t} - \eta^2 V^{k}_{t} \eta} \right]$$

guarantees for each subset $\mathcal{K}$ of experts, at each time $T \geq 0$:

$$R^{\mathcal{K}}_{T} \prec \sqrt{V^{\mathcal{K}}_{T} (- \ln \pi(\mathcal{K}))}$$

- Run-time of Hedge
Summary

- Hedge (robust tuning)
- Hedge (ad-hoc tuning)

Diagram shows regret vs. problem instances, with a minimax value and high (bad) values indicated.
Summary

- Hedge (robust tuning)
- Hedge (ad-hoc tuning)
- Squint

The graph illustrates the regret vs. problem instances for different tuning strategies. The regret is measured along the y-axis, ranging from minimax to high (bad), with 0 (perfect) at the bottom. The x-axis represents the problem instances.
Conclusion

Fresh algorithm for fundamental learning task
▶ new “different” perspective
▶ same efficiency
▶ adaptive (better) guarantees

Currently scaling up to advanced learning tasks
▶ Combinatorial games
▶ Matrix games
▶ Online optimization (gradient descent)

▶ Very welcome to discuss further
▶ Try it out

http://bitbucket.org/wmkoolen/squint
Thank you!