Follow the leader if you can,
Hedge if you must

Tim van Erven

NIPS, 2013

Joint work with: Steven de Rooij
Peter Grünwald
Wouter Koolen
Outline

• **Follow-the-Leader:**
  - works well for `easy' data: few leader changes, i.i.d.
  - but not robust to worst-case data

• **Exponential weights with simple tuning:**
  - robust, but does not exploit easy data

• **Second-order bounds:**
  - robust against worst case + can exploit i.i.d. data
  - but do not exploit few leader changes in general

• **FlipFlop:** **robust + as good as FTL**
Sequential Prediction with Expert Advice

- $K$ experts sequentially predict data $x_1, x_2, \ldots$
- Goal: predict (almost) as well as the best expert on average
- Applications:
  - online convex optimization
  - predicting electricity consumption
  - predicting air pollution levels
  - spam detection
  - ...

Set-up: Repeated Game

- Every round \( t = 1, \ldots, T \):
  1. Predict probability distribution
     \( w_t = (w_{t,1}, \ldots, w_{t,K}) \) on experts
  2. Observe expert losses
     \( \ell_t = (\ell_{t,1}, \ldots, \ell_{t,K}) \in [0,1]^K \)
  3. Our loss is
     \[ w_t \cdot \ell_t = \sum_k w_{t,k} \ell_{t,k} \]

Goal: minimize regret

\[
\sum_{t=1}^{T} w_t \cdot \ell_t - L^* \quad \text{where} \quad L^* = \min_k \sum_{t=1}^{T} \ell_{t,k}
\]
Follow-the-Leader

- Deterministically choose the expert that has predicted best in the past:

\[ \omega_{t,k^*} = 1 \text{ where } k^* = \arg \min_k \sum_{s=1}^{t-1} \ell_{t,k} \]

- Equivalently:

\[ \omega_t = \arg \min_w \mathbb{E}_{k \sim w} \left[ \sum_{s=1}^{t-1} \ell_{t,k} \right] \]
FTL: the Good News

- Regret bounded by nr of leader changes
- Proof sketch:
  - If the leader does not change, our loss is the same as the loss of the leader, so the regret stays the same
  - If the leader does change, our regret increases at most by 1 (range of losses)
- Works well for i.i.d. losses, because the leader changes only finitely many times w.h.p.
FTL on IID Losses

- 4 experts with Bernoulli 0.1, 0.2, 0.3, 0.4 losses
FTL Worst-case Losses

![Graph showing worst-case losses](image-url)
Exponential Weights

- **Follow-the-Leader:**
  \[ w_t = \arg \min_w \mathbb{E}_{k \sim w} \left[ \sum_{s=1}^{t-1} \ell_{t,k} \right] \]

- **Exponential weights:** add KL divergence from uniform distribution as a regularizer
  \[ w_t = \arg \min_w \mathbb{E}_{k \sim w} \left[ \sum_{s=1}^{t-1} \ell_{t,k} \right] + \frac{1}{\eta} D(w\|u) \]

- \( \eta \to \infty \): recover FTL (aggressive learning)

- As \( \eta \) closer to 0: closer to uniform distribution (more conservative learning)
Simple Tuning: the Good News

- Worst-case optimal for $\eta = \sqrt{8 \ln(K)/T}$:

$$\text{Regret} \leq \sqrt{T \ln(K)/2}$$

- Proof idea:
  - approximate our loss: $w_t \cdot l_t = \sum_k w_{t,k} l_{t,k}$
  - by the mix loss:
    $$m_t = \frac{-1}{\eta} \ln \sum_k w_{t,k} e^{-\eta l_{t,k}}$$
  - and bound the approximation error:
    $$\delta_t = w_t \cdot l_t - m_t$$
Simple Tuning: the Good News

\[
\text{our loss = mix loss + approx. error}
\]
\[
w_t \cdot \ell_t = m_t + \delta_t
\]

- Cumulative mix loss is close to \( L^* \):
  \[
  L^* \leq \sum_{t=1}^{T} m_t \leq L^* + \frac{\ln K}{\eta}
  \]

- Hoeffding's bound:
  \[
  \delta_t \leq \frac{\eta}{8}
  \]

- Together:
  \[
  \sum_{t=1}^{T} w_t \cdot \ell_t - L^* \leq \frac{\ln K}{\eta} + \frac{\eta T}{8}
  \]

\[
\eta = \sqrt{8 \ln K / T}
\]

\[
\sqrt{T \ln(K) / 2}
\]
Lost Advantages of FTL

- Simple tuning does much worse than FTL on i.i.d. losses
The bad news:

- $\eta = \sqrt{\frac{8 \ln(K)}{T}} = \text{conservative learning}$
- In practice, better when learning rate does not go to 0 with $T$! [DGGS, 2013]
- Lost advantages of FTL!

We want to exploit **luckiness**:

- robust against worst-case losses; but
- if the data are `easy', we should learn faster!
Luckiness: Exploiting Easy Data

- Improvement for small losses:

  \[ \text{Regret} = O\left(\sqrt{L^* \ln(K)}\right) \]

- Second-order Bounds:
  - [CBMS, 2007] and AdaHedge: 
    \[ O\left(\sqrt{\sum_t v_t \ln(K)}\right) \]
  - Related bound by [HK, 2008]

\[ w_t \] variance of
Luckiness: Exploiting Easy Data

- Improvement for small losses:
  \[
  \text{Regret} = O\left(\sqrt{L^* \ln(K)}\right)
  \]

- Second-order Bounds:
  - [CBMS, 2007] and AdaHedge:
    \[
    O\left(\sqrt{\sum_t v_t \ln(K)}\right)
    \]
  - Related bound by [HK, 2008]
    \[
    O\left(\sqrt{\frac{L^*(T - L^*)}{T} \ln(K)}\right)
    \]
The 2\textsuperscript{nd}-order Bounds: I.I.D. Data

- Regret bound: \( O\left(\sqrt{\sum_t v_t \ln(K)}\right) \)

- For IID data, \( w_t \) concentrates fast on best expert:
  \[
  \sum_t v_t \leq C \quad \Rightarrow \quad \text{Regret} \leq C'
  \]
2^{nd}\text{-}order Bounds: I.I.D. Data

Recover FTL benefits for i.i.d. data
CBMS: Proof Idea

\[
\text{our loss} = \text{mix loss} + \text{approx. error} \\
w_t \cdot \ell_t = m_t + \delta_t
\]

- Cumulative mix loss is close to \( L^* \):
  \[
  L^* \leq \sum_{t=1}^{T} m_t \leq L^* + \frac{\ln K}{\eta}
  \]

- Bernstein's bound:
  \[
  \delta_t \leq \frac{1}{2} \eta v_t + \text{lower order terms}
  \]

- Together:
  \[
  \text{Regret} \leq \frac{\ln K}{\eta} + \frac{1}{2} \eta \sum_{t=1}^{T} v_t \quad \xrightarrow{\text{balancing}} \quad O\left(\sqrt{\sum_t v_t \ln(K)}\right)
  \]
AdaHedge: Proof Idea

our loss = mix loss + approx. error
\[ w_t \cdot \ell_t = m_t + \delta_t \]

• Cumulative mix loss is close to \( L^* \):
\[ L^* \leq \sum_{t=1}^{T} m_t \leq L^* + \frac{\ln K}{\eta} \]

• No bound:
\[ \delta_t = \delta_t \]

• Together: balancing
\[ \eta = \frac{\ln(K)}{\sum_t \delta_t} \]
\[ \text{Regret} \leq \frac{\ln K}{\eta} + \sum_t \delta_t \rightarrow O\left(\sum_t \delta_t\right) = O\left(\sqrt{\sum_t v_t \ln K}\right) \]
AdaHedge: Proof Idea

- Cumulative mix loss is close to $L^*$:

$$L^* \leq \sum_{t=1}^{T} m_t \leq L^* + \frac{\ln K}{\eta}$$

- No bound:

$$\delta_t = \delta_t$$

- Together: balancing

$$\eta = \frac{\ln(K)}{\sum_t \delta_t}$$

$$\text{Regret} \leq \frac{\ln K}{\eta} + \sum_t \delta_t \rightarrow O\left(\sum_t \delta_t\right) = O\left(\sqrt{\sum_t v_t \ln K}\right)$$

NB Bernstein's bound is pretty sharp, so in practice CBMS $\approx$ AdaHedge up to constants.
Balancing $\eta$ in CBMS and AdaHedge depends on unknown quantities. Solve this by changing $\eta = \eta_t$ with $t$. Problem: $\sum_t m_t \leq L^* + \ln K/\eta$ breaks.

Lemma [KV, 2005]: If $\eta_1 \geq \eta_2 \geq \eta_3 \geq \ldots$, then
$$\sum_{t=1}^{T} m_t \leq L^* + \ln(K)/\eta_T$$
2nd-order Bounds: the Bad News

- Do not recover FTL benefits for other `easy' data with a small number of leader changes
Luckiness: Exploiting Easy Data

- Improvement for small losses:
  \[ \text{Regret} = O\left(\sqrt{L^* \ln(K)}\right) \]

- Second-order Bounds:
  - [CBMS, 2007] and AdaHedge: \[ O\left(\sqrt{\sum_t \nu_t \ln(K)}\right) \]
  - Related bound by [HK, 2008]

- FlipFlop:
  - “Follow the leader if you can, Hedge if you must”
  - Regret \( \leq \) best of AdaHedge and FTL
FlipFlop

- FlipFlop bound:

\[ \text{Regret} \leq \left\{ \begin{array}{ll}
6 \cdot \text{FTL Regret} \\
3 \cdot \text{AdaHedge Regret Bound}
\end{array} \right. \]

- Alternate Flip and Flop regimes
  - Flip: Tune $\eta_t = \infty$ like FTL
  - Flop: Tune $\eta_t$ like AdaHedge

- (No restarts of the algorithm, like in `doubling trick'!)
FlipFlop: Proof Ideas

• Alternate Flip and Flop regimes
  - Flip: Tune $\eta_t = \infty$ like FTL
  - Flop: Tune $\eta_t$ like AdaHedge

• Analysing two regimes:
  1. Relate mix loss for Flip to mix loss for Flop
  2. Keep approximation errors balanced between regimes
1. Relating Mix Losses

- We violate condition of KV-lemma:

\[ \eta_1 \geq \eta_2 \geq \eta_3 \geq \ldots \]

- But:

\[
\sum_t m_t \leq \sum_t m_t^{\text{flop}} + C \sum_{t \in \text{flop}} \delta_t \\
\leq L^* + \frac{\ln K}{\eta_T} + C \sum_{t \in \text{flop}} \delta_t \\
= L^* + (C + 1) \sum_{t \in \text{flop}} \delta_t
\]
2. Balance Approximation Errors

• Alternate regimes to keep approximation errors balanced:

\[
\sum_{t \in \text{flip}} \delta_t \propto \sum_{t \in \text{flop}} \delta_t
\]

\[
\text{Regret} = \sum_{t} m_t - L^* + \sum_{t} \delta_t \leq (C + 2) \sum_{t \in \text{flop}} \delta_t + \sum_{t \in \text{flip}} \delta_t
\]

\[
\propto \begin{cases} 
\sum_{t \in \text{flip}} \sum_{t} \delta_t & \rightarrow \text{FTL Bound} \\
\sum_{t \in \text{flop}} \sum_{t} \delta_t & \rightarrow \text{AdaHedge Bound}
\end{cases}
\]
Small Nr Leader Changes Again

- FlipFlop exploits easy data, AdaHedge does not
FTL Worst-case Again

![Graph showing regret vs. number of trials T]

- FTL
- AdaHedge
- FlipFlop
Summary

• **Follow-the-Leader:**
  - works well for *`easy' data*: i.i.d., few leader changes
  - but not robust to worst-case data

• **Second-order bounds** (e.g. CBMS, AdaHedge):
  - robust against **worst case** + can exploit **i.i.d. data**
  - but do not exploit few leader changes in general

• **FlipFlop:** **best of both worlds**
Luckiness: What's Missing?

- **FlipFlop:**
  - “Follow the leader if you can, Hedge if you must”
  - Regret $\leq$ best of AdaHedge and FTL

- But what if optimal $\eta$ is in between AdaHedge and FTL?
- Can we compete with the best possible $\eta$ chosen in hindsight?
References

- De Rooij, Van Erven, Grünwald, Koolen. *Follow the Leader If You Can, Hedge If You Must*. Accepted by the Journal of Machine Learning Research, 2013.
EXTRA SLIDES
No Need to Pre-process Losses

- Common assumption $\ell_{t,k} \in [0, 1]$ requires **translating and rescaling** the losses.

- **CBMS:**
  - Extension so this is **not necessary.** Important when range of losses is unknown!

- **AdaHedge and FlipFlop:**
  - Invariant under rescaling and translation of losses, so get this **for free.**
2\textsuperscript{nd}-order Bounds: I.I.D. Data

- Regret bound: \( O\left(\sqrt{\sum_t v_t \ln(K)}\right) \)
- If \( w_t \) concentrates fast on best expert, then
  \[
  \sum_t v_t \leq C \quad \Rightarrow \quad \text{Regret} \leq C'
  \]

- IID data:
  1. Balancing \( \eta_t = \sqrt{\frac{2 \ln(K)}{\sum_{s=1}^{t-1} v_s}} \) is large for all \( t \leq T \)
  2. \( w_t \) concentrates fast
  3. Then 1. also holds for \( T + 1 \)

\( \text{variance of } w_t \)
FlipFlop on I.I.D. Data

![Graph showing the performance of different algorithms over the number of trials T. The x-axis represents the number of trials T, ranging from 0 to 1000, and the y-axis represents the regret. Different algorithms are represented by distinct lines: FTL (blue), Simple tuning (green), AdaHedge (red), and FlipFlop (cyan). The graph illustrates the comparison of these algorithms over the given range of trials.]
Example: Spam Detection

<table>
<thead>
<tr>
<th>Subject</th>
<th>From</th>
<th>$y_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gratis Turkije. . .</td>
<td>Reizen Center</td>
<td>$y_1 = 1$</td>
</tr>
<tr>
<td>uitnodiging hoorzitting reorganisatie FEW dinsdag 20 se...</td>
<td>Ivo van Stokkum</td>
<td>$y_2 = 0$</td>
</tr>
<tr>
<td>Re: Urgent Business Inquiry.</td>
<td>Ubc Ltd</td>
<td>$y_3 = 1$</td>
</tr>
<tr>
<td>Reminder: first colloquium</td>
<td>Jeu, R.M.H. de</td>
<td>$y_4 = 0$</td>
</tr>
<tr>
<td>Informatie over VUnet</td>
<td>College van Bestuur</td>
<td>$y_5 = 0$</td>
</tr>
<tr>
<td>USD 500 Free Deposit at PartyPoker!</td>
<td>PartyPoker</td>
<td>$y_6 = 1$</td>
</tr>
<tr>
<td>YOU ARE A WINNER!!! VERY URGENT NOTIFICATION.</td>
<td>UK INTL. LOTTERY PROMOTION</td>
<td>$y_7 = 1$</td>
</tr>
<tr>
<td>bachelor/master diploma uitreiking 14 september</td>
<td>Sotiriou, M.</td>
<td>$y_8 = 0$</td>
</tr>
<tr>
<td>HAPPY NEW YEAR 2068</td>
<td>Anil Shilpakar</td>
<td>$y_9 = 1$</td>
</tr>
<tr>
<td>Thailand Package</td>
<td>Anil Shilpakar</td>
<td>$y_{10} = 1$</td>
</tr>
</tbody>
</table>
Example: Spam Detection

- **Data:** $(x_t, y_t)$ with $y_t \in \{0, 1\}$
- **Predictions:** probability $p_t \in [0, 1]$ that $y_t = 1$
- **Loss (probability of wrong label):**
  \[
  \ell(y_t, p_t) = \begin{cases} 
  p_t & \text{if } y_t = 0 \\
  1 - p_t & \text{if } y_t = 1 
  \end{cases}
  \]
- **Experts:** $K$ spam detection algorithms
- **If expert $k$ predicts $p_{t,k}$, then** $\ell_{t,k} = \ell(y_t, p_{t,k})$
- **Regret:** expected nr. mistakes over expected nr. of mistakes of best algorithm
FTL: the Bad News

- Consider two trivial spam detectors (experts):
  \[ p_{t,1} = 0 \quad p_{t,2} = 1 \]
- If we deterministically choose an expert \( k^* \) (like FTL) then we could be wrong all the time:
  \[ \ell_{t,k^*} = 1 \quad \ell_{t,-k^*} = 0 \]

Regret:
- Let \( n \) denote the number of times expert 1 has loss 1. Then \( L^* = \min\{n, T - n\} \leq T/2 \)
- **Linear regret** = \( T - L^* \geq T/2 \)