A Closer Look at Adaptive Regret

Dmitry Adamskiy **Wouter M. Koolen** Volodya Vovk
Alexey Chernov

Computer Learning Research Centre
Department of Computer Science
Royal Holloway University of London

14/02/2013

Outline

1. Why adaptive regret?
2. Setup
3. Results

Weather forecasting: adaptivity

Predictor 

Expert

Expert

Expert

Nature 

Weather forecasting: adaptivity

Predictor		
Expert		30%
Expert		90%
Expert		20%
Nature		

Weather forecasting: adaptivity

Predictor		55%
Expert	france 2	30%
Expert	TF1	90%
Expert	CANAL+	20%
Nature		

Weather forecasting: adaptivity

Predictor		55%
Expert	france 2	30%
Expert	TF1	90%
Expert	CANAL+	20%
Nature		

Weather forecasting: adaptivity

Predictor		55%	
Expert		30%	40%
Expert		90%	70%
Expert		20%	65%
Nature			

Weather forecasting: adaptivity

Predictor		55%	65%
Expert		30%	40%
Expert		90%	70%
Expert		20%	65%
Nature			

Weather forecasting: adaptivity

Predictor		55%	65%
Expert	france 2	30%	40%
Expert	TF1	90%	70%
Expert	CANAL+	20%	65%
Nature			

Weather forecasting: adaptivity

Predictor		55%	65%	
Expert	france 2	30%	40%	...
Expert	TF1	90%	70%	...
Expert	CANAL+	20%	65%	...
Nature				

Weather forecasting: adaptivity

Predictor		55%	65%	...
Expert	france 2	30%	40%	...
Expert	TF1	90%	70%	...
Expert	CANAL+	20%	65%	...
Nature				

Weather forecasting: adaptivity

Predictor		55%	65%	...
Expert		30%	40%	...
Expert		90%	70%	...
Expert		20%	65%	...
Nature				...

Goal: close to the best expert overall (solution: AA)

Weather forecasting: adaptivity

Predictor		55%	65%	...	
Expert	france 2	30%	40%	...	is bad on foggy days!
Expert	TF1	90%	70%	...	
Expert	CANAL+	20%	65%	...	
Nature				...	

Goal: close to the best expert overall (solution: AA)

Weather forecasting: adaptivity

Predictor		55%	65%	…	
Expert	france 2	30%	40%	…	is bad on foggy days!
Expert	TF1	90%	70%	…	
Expert	CANAL+	20%	65%	…	drunk on weekends!
Nature				…	

Goal: close to the best expert overall (solution: AA)

Weather forecasting: adaptivity

Predictor		55%	65%	…	
Expert	france 2	30%	40%	…	is bad on foggy days!
Expert	TF1	90%	70%	…	goes on training!
Expert	CANAL+	20%	65%	…	drunk on weekends!
Nature				…	

Goal: close to the best expert overall (solution: AA)

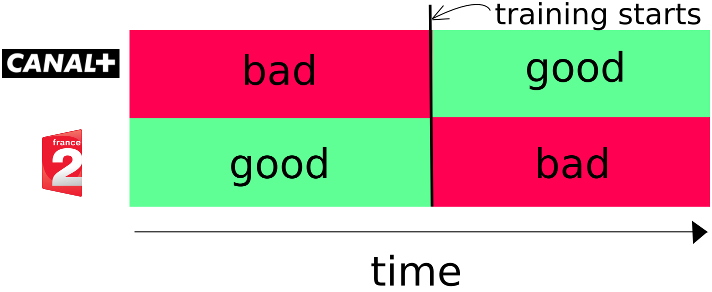
Weather forecasting: adaptivity

Predictor		55%	65%	…	
Expert	france 2	30%	40%	…	is bad on foggy days!
Expert	TF1	90%	70%	…	goes on training!
Expert	CANAL+	20%	65%	…	drunk on weekends!
Nature				…	

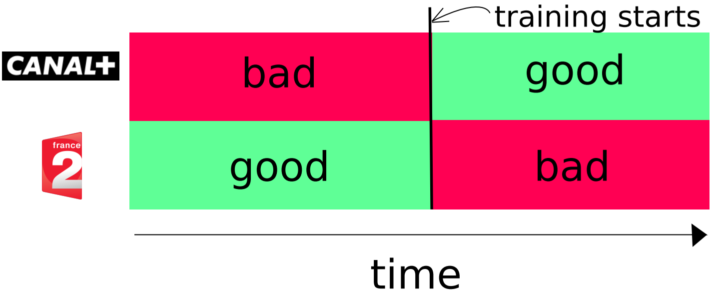
Goal: close to the best expert overall (solution: AA)

Adaptive goal: close to the best expert on every time interval

Example continued



Example continued



Non-adaptive predictor would lose trust in the first guy.

Adaptive algorithms

We studied several approaches to adaptivity:

- Blowing up the set of experts to compete with virtual sleeping experts [Adamskiy, Koolen, Chernov and Vovk 2012]

Adaptive algorithms

We studied several approaches to adaptivity:

- Blowing up the set of experts to compete with virtual sleeping experts [Adamskiy, Koolen, Chernov and Vovk 2012]
  Turned out to be Fixed Share [Herbster and Warmuth 1998]!

Adaptive algorithms

We studied several approaches to adaptivity:

- Blowing up the set of experts to compete with virtual sleeping experts [Adamskiy, Koolen, Chernov and Vovk 2012]
  Turned out to be Fixed Share [Herbster and Warmuth 1998]!
- Restarting existing algorithms and combining their predictions [Hazan and Seshadhri 2009]

Adaptive algorithms

We studied several approaches to adaptivity:

- Blowing up the set of experts to compete with virtual sleeping experts [Adamskiy, Koolen, Chernov and Vovk 2012]
  Turned out to be Fixed Share [Herbster and Warmuth 1998]!
- Restarting existing algorithms and combining their predictions [Hazan and Seshadhri 2009]
  Also turned out to be Fixed Share!

Adaptive properties of Fixed Share: results

Fixed Share is known for *tracking*. What about its adaptivity?

Our results

1. Figured out the worst-case adaptive regret of Fixed Share

Adaptive properties of Fixed Share: results

Fixed Share is known for *tracking*. What about its adaptivity?

Our results

1. Figured out the worst-case adaptive regret of Fixed Share
2. Proved the optimality of Fixed Share — every algorithm is dominated by a Fixed Share (i.t.o. worst-case adaptive regret)

1. Why adaptive regret?
2. **Setup**
3. Results

Protocol: Mix loss

for $t = 1, 2, \dots$ **do**

Learner announces probability vector $\boldsymbol{w}_t \in \triangle_N$

Reality announces loss vector $\boldsymbol{\ell}_t \in [-\infty, \infty]^N$

Learner suffers loss $\ell_t := -\ln \sum_n w_t^n e^{-\ell_t^n}$

end for

Adaptive Regret

- Goal: On every time interval $[t_1, t_2]$ we want to be not much worse than the best expert on that interval.
- We are interested in small adaptive regret

Adaptive Regret

- Goal: On every time interval $[t_1, t_2]$ we want to be not much worse than the best expert on that interval.
- We are interested in small adaptive regret

Definition

The adaptive regret of the algorithm on the interval $[t_1, t_2]$ is the loss of the algorithm there minus the lost of the best expert there:

$$R_{[t_1,t_2]} := L_{[t_1,t_2]} - \min_j L^j_{[t_1,t_2]}$$

Adaptive Regret

- Goal: On every time interval $[t_1, t_2]$ we want to be not much worse than the best expert on that interval.
- We are interested in small adaptive regret

Definition

The adaptive regret of the algorithm on the interval $[t_1, t_2]$ is the loss of the algorithm there minus the lost of the best expert there:

$$R_{[t_1,t_2]} := L_{[t_1,t_2]} - \min_j L^j_{[t_1,t_2]}$$

- We get worst-case adaptive regret by maximising over the choice of expert losses.

AA and Fixed Share

Aggregating Algorithm [Vovk 1990] updates weights as:

$$w_{t+1}^n := \frac{w_t^n e^{-\ell_t^n}}{\sum_n w_t^n e^{-\ell_t^n}}.$$

AA and Fixed Share

Aggregating Algorithm [Vovk 1990] updates weights as:

$$w_{t+1}^n \;:=\; \frac{w_t^n e^{-\ell_t^n}}{\sum_n w_t^n e^{-\ell_t^n}}.$$

Fixed Share (family) is defined by a sequence of "switching rates" α_t. The weight update is

$$w_{t+1}^n \;:=\; \frac{\alpha_{t+1}}{N-1} + \left(1 - \frac{N}{N-1}\alpha_{t+1}\right) \frac{w_t^n e^{-\ell_t^n}}{\sum_n w_t^n e^{-\ell_t^n}}.$$

AA and Fixed Share

Aggregating Algorithm [Vovk 1990] updates weights as:

$$w_{t+1}^n := \frac{w_t^n e^{-\ell_t^n}}{\sum_n w_t^n e^{-\ell_t^n}}.$$

Fixed Share (family) is defined by a sequence of “switching rates” α_t. The weight update is

$$w_{t+1}^n := \frac{\alpha_{t+1}}{N-1} + \left(1 - \frac{N}{N-1}\alpha_{t+1}\right) \frac{w_t^n e^{-\ell_t^n}}{\sum_n w_t^n e^{-\ell_t^n}}.$$

Adaptivity hides in the first term.

Fixed Share Worst-Case Adaptive regret data

We proved that the worst case data for Fixed Share looks like this:

	1	...	$t_1 - 1$	t_1	...	t_2
Expert 1	?	?		0	0	0
Expert 2	?	?	?			
...	?	?	?			
Expert N	?	?	?			

where  denotes infinite loss, 0 – zero loss and '?' – losses that don't matter.

Fixed Share Worst-Case Adaptive regret formula

Knowing the worst-case data, we can plug it in and calculate the regret:

Theorem

The worst-case adaptive regret of Fixed Share with N experts on interval $[t_1, t_2]$ equals

$$-\ln\left(\frac{\alpha_{t_1}}{N-1}\prod_{t=t_1+1}^{t_2}(1-\alpha_t)\right).$$

Different α-s: examples

- Classic Fixed Share ($\alpha_t = const$):

$$\begin{array}{ll} \ln(N-1) - \ln\alpha - (t_2 - t_1)\ln\bigl(1-\alpha\bigr) & \text{for } t_1 > 1, \text{ and} \\ \ln N - (t_2 - 1)\ln\bigl(1-\alpha\bigr) & \text{for } t_1 = 1. \end{array}$$

Linear in length $t_2 - t_1$

Different α-s: examples

- Classic Fixed Share ($\alpha_t = const$):

$$\begin{aligned} &\ln(N-1) - \ln\alpha - (t_2 - t_1)\ln\big(1-\alpha\big) && \text{for } t_1 > 1, \text{ and} \\ &\ln N - (t_2 - 1)\ln\big(1-\alpha\big) && \text{for } t_1 = 1. \end{aligned}$$

Linear in length $t_2 - t_1$

- Slowly decreasing $\alpha_t = 1/t$ leads to regret of

$$\begin{aligned} &\ln(N-1) + \ln t_2 && \text{for } t_1 > 1, \text{ and} \\ &\ln N + \ln t_2 && \text{for } t_1 = 1. \end{aligned}$$

Logarithmic in end-time t_2

Different α-s: examples

- Quickly decreasing switching rate.
  If we set $\alpha_t = t^{-2}$ we have the upper bound for regret

  $$\ln N + 2 \ln t_1 + \ln 2.$$

  Logarithmic in start-time t_1

  For $t_1 = 1$ this is very close to classical AA regret!

Combining adaptive stuff to get tracking bounds

The worst-case data we have just shown could be combined over the intervals, thus giving the overall worst-case data for partitions.

Combining adaptive stuff to get tracking bounds

The worst-case data we have just shown could be combined over the intervals, thus giving the overall worst-case data for partitions.

Time	Interval 1				Interval 2				Interval 3			
Expert 1	0	0	...	0	☠	☠	...	☠	☠	☠	...	☠
Expert 2	☠	☠	...	☠	0	0	...	0	☠	☠	...	☠
Expert 3	☠	☠	...	☠	☠	☠	...	☠	0	0	...	0

Combining adaptive stuff to get tracking bounds

The worst-case data we have just shown could be combined over the intervals, thus giving the overall worst-case data for partitions.

Time	Interval 1				Interval 2				Interval 3			
Expert 1	0	0	...	0	☠	☠	...	☠	☠	☠	...	☠
Expert 2	☠	☠	...	☠	0	0	...	0	☠	☠	...	☠
Expert 3	☠	☠	...	☠	☠	☠	...	☠	0	0	...	0

And the tracking bound can be recovered!

Optimality

Ok, we have a regret of *some* algorithm. But is it optimal?

Optimality

Ok, we have a regret of *some* algorithm. But is it optimal?
It turns out to be!

Optimality

Ok, we have a regret of *some* algorithm. But is it optimal?
It turns out to be!

Theorem

The worst-case adaptive regret of any algorithm is dominated by that of a Fixed Share.

Optimality

Theorem (Lower bound)

Let N be the number of experts. If $\phi(t_1, t_2)$ is the worst-case adaptive regret of an algorithm, then

$$\phi(t, t) \geq \ln N \qquad \textit{for all } 1 \leq t \textit{ and}$$

$$\phi(t_1, t_2) \geq \phi(t_1, t_1) - \sum_{t=t_1+1}^{t_2} \ln\left(1 - (N-1)e^{-\phi(t,t)}\right) \qquad \textit{for all } 1 \leq t_1 < t_2.$$

Optimality

Theorem (Lower bound)

Let N be the number of experts. If $\phi(t_1, t_2)$ is the worst-case adaptive regret of an algorithm, then

$$\phi(t,t) \geq \ln N \qquad \text{for all } 1 \leq t \text{ and}$$

$$\phi(t_1, t_2) \geq \phi(t_1, t_1) - \sum_{t=t_1+1}^{t_2} \ln\left(1 - (N-1)e^{-\phi(t,t)}\right) \qquad \text{for all } 1 \leq t_1 < t_2.$$

Fixed Share with

$$\alpha_t = (N-1)e^{-\phi(t,t)}.$$

will have the second line with equality.

Summary

- We studied two intuitive methods to obtain adaptive algorithms.
- They turned out to be Fixed Share.
- We studied the worst-case Adaptive Regret of Fixed Share.
- We showed that Fixed Share is optimal.

Thank you!