

Some comments on “A parameter-free hedging algorithm” by Chaudhuri, Freund, and Hsu

Volodya Vovk

November 23, 2009

This paper [3] makes (at least) two important contributions to the decision-theoretic framework for on-line prediction (DTOL), introduced by Freund and Schapire’s [4]:

- The bound for the proposed prediction algorithm works well when the “effective” number of actions is much smaller than the nominal number of actions.
- The actions that perform worse than the prediction algorithm are ignored by the algorithm.

As the authors say in Section 4, the second feature of their algorithm is shared by polynomial weights algorithms, but the first contribution is a real breakthrough.

The paper is very clearly written. I strongly recommend reading its first section to everybody interested in prediction with expert advice. The difference that the authors make between the “nominal” and “effective” numbers of actions (more generally, of experts) is important, and I do not remember seeing it clearly discussed in other papers.

Cesa-Bianchi and Lugosi’s result

The main result of [3] is Theorem 1 on the fourth page (one of my minor complaints about [3] is that the pages are not numbered). Perhaps the strongest known result of a similar flavour is Theorem 2.3 in Cesa-Bianchi and Lugosi’s book [2] (p. 16), discussed in [3], Section 4. However, Theorem 2.3 of [2] is not stated in the DTOL framework, and I will start this section by explaining connections between DTOL and the more general framework of prediction with expert advice (in principle, these connections are well known, but I will spell them out for the sake of readers from outside this area).

In the DTOL framework, the learner interacts with his environment in discrete time, $t = 1, 2, \dots$. In round t he chooses a weight distribution $(p_{i,t})_{i=1}^N$ on the set $\{1, \dots, N\}$ of N actions (so that $p_{1,t}, \dots, p_{N,t}$ are nonnegative numbers summing to 1). At the end of round t the learner is told the loss $l_{i,t} \in [0, 1]$ of

each action $i = 1, \dots, N$ in this round, and his loss is defined to be the mean value

$$l_{A,t} = \sum_{i=1}^N p_{i,t} l_{i,t}.$$

A typical result that we are interested in in the DTOL framework is that the learner can guarantee that, for all i ,

$$\sum_{t=1}^T l_{A,t} \leq \sum_{t=1}^T l_{i,t} + \sqrt{(1 + \ln N) (78T + 32 \ln^2 N (40.8 + \ln N))} \quad (1)$$

(this is a special case of the second part of Theorem 1 in [3]).

In the framework of prediction with expert advice (as described in, e.g., [2], p. 7), in each round t the learner has access to decisions (or predictions) $(\gamma_{i,t})_{i=1}^N$ made by N experts, is required to come up with his own prediction γ_t , and is told the outcome ω_t produced by the environment. The loss suffered by the learner and each expert is measured by a loss function $\lambda(\gamma, \omega)$, which will be assumed to take values in the interval $[0, 1]$.

An important special case is where the allowed predictions form a convex set in a linear space and the function $\lambda(\gamma, \omega)$ is convex in its first argument γ . It is very easy to see from (1) that in this case the learner can guarantee

$$\sum_{t=1}^T \lambda(\gamma_t, \omega_t) \leq \sum_{t=1}^T \lambda(\gamma_{i,t}, \omega_t) + \sqrt{(1 + \ln N) (78T + 32 \ln^2 N (40.8 + \ln N))} \quad (2)$$

for all i . Indeed, the learner can use the strategy guaranteeing (1): in round t he should just output any decision γ_t satisfying

$$\lambda(\gamma_t, \omega) \leq \sum_{i=1}^N p_{i,t} \lambda(\gamma_{i,t}, \omega)$$

for any ω ; by the convexity condition, $\gamma_t = \sum_{i=1}^N p_{i,t} \gamma_{i,t}$ will do.

On the other hand, if we know that (2) holds for all loss functions that are convex in the first argument, we can easily deduce that (1) holds in the DTOL framework. (Remember that loss functions are always assumed to take values in $[0, 1]$.) Indeed, the DTOL framework corresponds to the following special case of prediction with expert advice:

- the outcome ω_t is defined to be the vector $(l_{1,t}, \dots, l_{N,t})$ of the actions' losses;
- the learner's prediction γ_t is the vector $(p_{1,t}, \dots, p_{N,t})$;
- there are N experts, with expert $i = 1, \dots, N$ predicting $\gamma_{i,t} = (0, \dots, 0, 1, 0, \dots, 0)$ (the only 1 is at the i th position);
- the loss function $\lambda(\gamma, \omega)$ is the dot product $\gamma \cdot \omega$ of the prediction and the outcome.

We can see that results of the type (1) can be readily translated from the DTOL language to the language of prediction with expert advice, and vice versa. However, the equivalence between DTOL and prediction with expert advice is limited: there are many results in prediction with expert advice for specific loss functions (such as the absolute loss function: see, e.g., [2], Chapter 8), and there is no way to translate such results into DTOL. It appears that “specialization” would be a better term than “generalization” to use in the title of [4].

Theorem 2.3 in [2] (p. 17) says that in the case where the loss function is convex in the first argument the learner can guarantee

$$\sum_{t=1}^T \lambda(\gamma_t, \omega_t) \leq \sum_{t=1}^T \lambda(\gamma_{i,t}, \omega_t) + \sqrt{2T \ln N} + \sqrt{\frac{\ln N}{8}} \quad (3)$$

for all i . This implies that in the DTOL framework the learner can guarantee

$$\sum_{t=1}^T l_{A,t} \leq \sum_{t=1}^T l_{i,t} + \sqrt{2T \ln N} + \sqrt{\frac{\ln N}{8}} \quad (4)$$

for all i .

Chaudhuri et al. ([3], Section 4) cite [1] as the source of this result. This is a misunderstanding: Cesa-Bianchi and Lugosi say in [2] (p. 36) that the analysis of Theorem 2.3 is adapted from [1], not the theorem itself; I could not find anything like Cesa-Bianchi and Lugosi’s Theorem 2.3 in [1]. It appears that the algorithm achieving (4) was first explicitly described by Kalnishkan and Vyugin in [5] under the name of the “weak aggregating algorithm” (Cesa-Bianchi and Lugosi’s name is the “exponentially weighted average forecaster with time-varying potential”). Some of the ideas used in this algorithm go back to [1] (as it so often happens in learning theory, they were rediscovered independently by Kalnishkan and Vyugin). The bound given in [5] is somewhat weaker than (4) (unless $T = 0$); in the language of DTOL it can be written as

$$\sum_{t=1}^T l_{A,t} \leq \sum_{t=1}^T l_{i,t} + 2\sqrt{T \ln N}. \quad (5)$$

Competing with an unknown effective number of actions

The bounds (4) and (5) are significantly stronger than (1). However, (1) can be strengthened to

$$\sum_{t=1}^T l_{A,t} \leq \sum_{t=1}^T l_{i,t} + \sqrt{\left(1 + \ln \frac{1}{\epsilon}\right) (78T + 32 \ln^2 N (40.8 + \ln N))}, \quad (6)$$

where i is the $\lfloor \epsilon N \rfloor$ th best action, for any $\epsilon \in (0, 1)$ (this is a special case of the first part of Theorem 1 in [3]). The last inequality is the same as (1) for

$\epsilon = 1/N$, but the generalization to an arbitrary $\epsilon \in (0, 1)$ is very important: if, for example, there are only K “effective actions” and each effective action is copied by $N/K \gg 1$ “nominal actions”, we can take $\epsilon = 1/K$ rather than $\epsilon = 1/N$ (see [3], Section 1).

A natural question is whether the ability to cope with a small effective number of actions is a genuine advantage of Normal-Hedge over alternative algorithms; one might suspect that the authors of the alternative algorithms simply did not bother to write out their bounds in terms of ϵ rather than N . For example, the bounds for the weighted majority algorithm or the aggregating algorithm are usually stated in terms of N , but it is trivial to replace the term $\ln N$ by $\ln \frac{1}{\epsilon}$.

I believe that the advantage of Normal-Hedge is genuine. The argument in [2] (Theorem 2.3) is based on tracking the performance of the best nominal action, and I cannot see how it could be adapted to the case of a small (and unknown) number of effective actions. My understanding of the intuition behind Cesa-Bianchi and Lugosi’s argument is rather weak, but I understand Kalnishkan and Vyugin’s [5] argument much better (I even tried to rewrite it in the form that I find more intuitive: see [7]). Kalnishkan and Vyugin’s argument gives

$$\sum_{t=1}^T l_{A,t} \leq \sum_{t=1}^T l_{i,t} + c\sqrt{T} + \frac{1}{c}\sqrt{T} \ln \frac{1}{\epsilon},$$

where c is an *a priori* chosen constant and i , as before, the $\lfloor \epsilon N \rfloor$ th best action. If ϵ were known in advance, we could optimize c to obtain

$$\sum_{t=1}^T l_{A,t} \leq \sum_{t=1}^T l_{i,t} + 2\sqrt{T \ln \frac{1}{\epsilon}},$$

which would compare favourably with (6). However, as ϵ is unknown, we cannot get anything better than

$$\sum_{t=1}^T l_{A,t} \leq \sum_{t=1}^T l_{i,t} + O\left(\sqrt{T} \ln \frac{1}{\epsilon}\right),$$

which is significantly worse than (6) as $T \rightarrow \infty$ and $\epsilon \rightarrow 0$.

Comparison with the Hedge algorithm

There is another loss bound for the DTOL framework which is not comparable with either (4) or (6). Cesa-Bianchi and Lugosi ([2], Section 2.4) refer to this bound as an improvement for small losses. Freund and Schapire prove in [4] (Theorem 2) that their Hedge algorithm guarantees

$$\sum_{t=1}^T l_{A,t} \leq \frac{\eta \sum_{t=1}^T l_{i,t} + \ln N}{1 - e^{-\eta}} \tag{7}$$

in the DTOL framework, where $\eta > 0$ is the chosen “learning rate”. (Cesa-Bianchi and Lugosi [2] state this result as Theorem 2.4 and attribute it to Littlestone and Warmuth [6].) For $N \rightarrow \infty$ the loss bound (7) is optimal ([4], Theorem 3), but for finite N it can be improved to

$$\sum_{t=1}^T l_{A,t} \leq \frac{\eta \sum_{t=1}^T l_{i,t} + \ln N}{N \ln \frac{N}{N+e^{-\eta}-1}} \quad (8)$$

([8], Example 7). The last bound is obtained for the aggregating algorithm and it is easy to check that we can replace the $\ln N$ in the numerator of the right-hand side of (8) by $\ln \frac{1}{\epsilon}$. Therefore, we can also replace the $\ln N$ in (7) by $\ln \frac{1}{\epsilon}$ (this is part of Theorem 2 in [4]).

To summarize, at this time we have (at least) three incomparable results in the DTOL framework: (4), (6), and (8) (or (7) when N is large). Of these, only (6) and (8) can cope with a small number of effective actions.

Other remarks

There is an inaccuracy in the only displayed equation on the first page of the paper: $\ln^3 N$ should be multiplied by $\ln \frac{1}{\epsilon}$. This formula mentions T , whereas there is no T in Theorem 1 (and t is not even defined there). In lines 3 and 8 after this formula, $\ln^3 N$ should read $\ln^4 N$.

I found the description of the intuition behind Normal-Hedge in Section 3 too brief and too much intertwined with the BW algorithm. (The authors mention space constraints, but this is surely not applicable to the arXiv version.) I would prefer to see a self-contained explanation of the intuition behind Normal-Hedge only, perhaps with forward references to the actual proof. For example, the authors say that there is a potential function whose average value does not increase between iterations of the game; a reference to the corresponding place in the proof might be useful.

In Section 5, it would be interesting to see the comparison between Normal-Hedge and the Kalnishkan–Vyugin–Cesa-Bianchi–Lugosi algorithm with learning rate $\eta_t = 1/\sqrt{t}$. One interpretation of the pictures in this section is that the Exp and Poly algorithms are quite robust to misspecifying the effective number of actions; however, eventually they do become worse than Normal-Hedge.

References

- [1] Peter Auer, Nicolò Cesa-Bianchi, and Claudio Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64:48–75, 2002.
- [2] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, Cambridge, England, 2006.

- [3] Kamalika Chaudhuri, Yoav Freund, and Daniel Hsu. A parameter-free hedging algorithm. Technical Report [arXiv:0903.2851v1](#) [cs.LG], [arXiv.org](#) e-Print archive, March 2009.
- [4] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.
- [5] Yuri Kalnishkan and Michael V. Vyugin. The weak aggregating algorithm and weak mixability. In *COLT 2005*, pages 188–203. Journal version: *Journal of Computer and System Sciences* (COLT 2005 Special Issue) 74:1228–1244 (2008).
- [6] Nick Littlestone and Manfred K. Warmuth. The Weighted Majority Algorithm. *Information and Computation*, 108:212–261, 1994.
- [7] Vladimir Vovk. Competing with stationary prediction strategies. In *COLT 2007*, pages 293–307. Full version: Technical Report [arXiv:cs/0607067](#) [cs.LG], [arXiv.org](#) e-Print archive, March 2007.
- [8] Vladimir Vovk. A game of prediction with expert advice. *Journal of Computer and System Sciences*, 56:153–173, 1998.