Soft-Bayes: Prod for Mixtures of Experts with Log-Loss

Laurent Orseau and **Tor Lattimore** and **Shane Legg**
{LORSEAU,LATTIMORE,LEGG}@GOOGLE.COM
DeepMind, London, UK

Editors: Steve Hanneke and Lev Reyzin

Abstract

We consider prediction with expert advice under the log-loss with the goal of deriving efficient and robust algorithms. We argue that existing algorithms such as exponentiated gradient, online gradient descent and online Newton step do not adequately satisfy both requirements. Our main contribution is an analysis of the Prod algorithm that is robust to any data sequence and runs in linear time relative to the number of experts in each round. Despite the unbounded nature of the log-loss, we derive a bound that is independent of the largest loss and of the largest gradient, and depends only on the number of experts and the time horizon. Furthermore we give a Bayesian interpretation of Prod and adapt the algorithm to derive a tracking regret.

Keywords: Online convex optimization, logarithmic loss, prediction with expert advice.

1. Introduction

Sequence prediction is a simple problem at the core of many machine learning problems: Given a sequence of past observations, what is the probability of the next one? We approach this problem using the prediction with expert advice framework with logarithmic loss. The log-loss is arguably the most fundamental choice because of its connections to probability (minimising the log-loss is maximising the likelihood of the observed data), sequential betting, information theory and compression (minimising the log-loss corresponds to maximising the compression ratio when compressing using arithmetic coding).

Setup Let $\mathcal{X}$ be a countable alphabet and $\mathcal{D}$ be the set of probability distributions over $\mathcal{X}$. We consider a game over T rounds, where in each round t the learner first receives the predictions of N experts $(p_t^i)_{i=1}^N$ with $p_t^i \in \mathcal{D}$. The learner then chooses their own prediction $M_t \in \mathcal{D}$ and the next symbol $x_t \in \mathcal{X}$ is revealed. The learner then suffers an *instantaneous loss* at round t of

$$\ell_t(M_t) = -\log\left(M_t(x_t)\right)$$

and the cycles continue up to round T, which may or may not be known in advance. The learner would like to combine the advice of the experts so as to make the loss as small as possible, which corresponds to good prediction/compression. We make no assumptions on the data source; in particular $x_{1:T} = x_1, x_2, \ldots, x_T$ need not be independent and identically distributed, and may even be generated by an adversary. The standard approach in this setting is to analyse the regret of the predictor M relative to some interesting class of competitors. Here we focus on predicting well relative to a fixed convex combination of experts. Let $\mathcal{W}$ be the $(N-1)$-dimensional probability simplex: $\forall w \in \mathcal{W} : \sum_{i=1}^N w^i = 1$ and $\forall i \in [N] : w^i \in [0,1]$, where w^i is the ith component of the vector w. Define the *regret* (also called redundancy for the log-loss) relative to $a \in \mathcal{W}$ by

$$\mathcal{R}_T(a) = \sum_{t=1}^T \left[\ell_t(M_t) - \ell_t\left(\sum_{i=1}^N a^i p_t^i\right)\right] . \tag{1}$$

This definition of the regret is more demanding than the usual notion of competing with the best single expert in hindsight, which corresponds to competing with 'Dirac' experts $a \in \{e_1, e_2, \dots, e_N\}$ with e_i the standard basis vectors. In particular, defining the regret as in Eq. (1) forces the learner to exploit the 'wisdom of the crowd' by combining the experts predictions, rather than focusing on the single best expert in hindsight. This is often crucial because it often happens that no single expert predicts well over all time periods, and the nature of the log-loss means that even a single round of poor prediction can lead to a large instantaneous loss.

This framework has attracted significant attention over the last three decades, mainly due to its applications to compression and portfolio optimisation; see for example Kalai and Vempala (2002) and references within. Our objective is to design algorithms that are (a) linear-time in the number of experts N, (b) robust to the existence of incompetent experts and (c) recover the tracking guarantees of fixed share (Herbster and Warmuth, 1998). A variety of approaches are now known for this problem, but none satisfy all of (a), (b) and (c) above. The minimax optimal and Bayesian solutions are known to achieve logarithmic regret (Cover, 1991), but there are currently no linear-time implementations and it seems unlikely that one exists. The main computational challenge for the Bayesian approach is evaluating the normalisation integral. This issue was addressed by Kalai and Vempala (2002) via a polynomial-time sampling approach, but unfortunately the solution far from linear in N and is not suitable for practical implementations when N is large. More recently Hazan et al. (2007) proposed the online Newton step, a pseudo second-order algorithm that also achieves logarithmic regret, but depends on computing a generalised projection for which the best-known running time is $O(N^2)$ per step (Luo et al., 2016). Even in the idealised case that the projection could be computed in $O(1)$, the algorithm depends on maintaining and updating a covariance matrix and so $O(N^2)$ running time is unavoidable without some form of dimensionality reduction that weakens the regret guarantees (see Luo et al., 2016).

All of the algorithms mentioned until now enjoy optimal logarithmic regret. The exponentiated gradient (EG) algorithm by Helmbold et al. (1998) has a regret bound that looks like

$$\mathcal{R}_T(a) \leq C\sqrt{\frac{T}{2}\log N} , \tag{2}$$

where $C = \max_{t,i} \frac{p_t^i(x_t)}{M_t(x_t)} \leq \max_t \frac{1}{M_t(x_t)}$. The EG algorithm runs in $O(N)$ time per round, but unfortunately its regret depends on C, which may be so large that it can make Eq. (2) vacuous (Section 3). Even worse, the dependence on C is not an artifact of the analysis, but rather a failing of the EG algorithm, which becomes unstable when experts transition from predicting badly to predicting well. Another algorithm with near-linear running time is online gradient descent (OGD) by Zinkevich (2003) (and applied to this setting by Veness et al. (2012b)), which runs in $O(N\log(N))$ time using the fast simplex projection by Duchi et al. (2008). The regret of this algorithm also depends on the size of the maximum gradient of the loss, however, which leads to bound of the same order as Eq. (2). Note that EG is equivalent to using mirror descent with neg-entropy regularisation, which makes the projection to the simplex nothing more than normalisation. We will return briefly to alternative regularisation choices for mirror descent in the discussion.

We revisit the Prod algorithm by Cesa-Bianchi et al. (2007), which runs in $O(N)$ time per step and was originally designed for obtaining second-order bounds when competing with the best expert rather than the mixture. The stability of Prod has not gone unnoticed, with recent work by Gaillard et al. (2014) and Sani et al. (2014) also exploiting its advantages over exponential weighting. Since the log-loss is unbounded, it is not immediately suitable for use in Prod, but conveniently the linearised loss *is* semi-bounded. In this sense our algorithm is to Prod what exponentiated gradient is to exponential weighting.

Contributions Our main contribution is an analysis of Prod when competing against a mixture in the log-loss setting (Sections 4 and 5). By tuning the learning rate we are able to show two regret bounds:

$$\mathcal{R}_T = O\left(\sqrt{NT\log N}\right) \qquad (3) \qquad\qquad \mathcal{R}_T = O\left(\sqrt{TC\log N}\right)\,, \qquad (4)$$

where C is defined as above. The first bound (Section 4) eliminates *all dependence* on the arbitrarily large C at the price of a square-root dependence on the dimension N. The second bound (Section 5) retains a dependence on C, but moves it inside the square root relative to the EG algorithm. We also prove self-confident bounds (Section 5) and analyse a truly online version of Prod that does not need prior knowledge of the horizon and simultaneously achieves a tracking guarantee (Section 6). To complement the upper bounds we prove lower bounds for EG and OGD showing that in the worst case they suffer nearly *linear* regret (Section 3). Moreover, we give two Bayesian interpretations of Prod as a 'slowed down' Bayesian predictor or a mixture of 'partially sleeping' experts (Section 2).

Notation For a natural number n let $[n] = \{1, 2, \dots, n\}$ and e_i be the standard basis vectors (the dimension will always be clear from context). The number of experts is denoted by $N \geq 1$ and the time horizon is $T \geq 1$. Let $\mathcal{D}$ be the set of probability distributions on the countable alphabet $\mathcal{X}$, and $\mathcal{W}$ be the $(N-1)$-dimensional probability simplex so that for $a \in \mathcal{W}$ we have $a^i \geq 0$ for all i and $\sum_{i=1}^N a^i = 1$. For $a, w \in \mathcal{W}$ we define $\mathrm{RE}(a\|b) = \sum_{i:a^i>0}^N a^i \ln(a^i/b^i)$ to be the relative entropy between a and b. All of the following analysis only relies on p_t^i through $p_t^i(x_t)$, which is the probability that expert i assigned to the actual observation x_t in round t. For this reason we abbreviate $p_t^i = p_t^i(x_t)$. We also use $p_{1:t}^i = \prod_{s=1}^t p_s^i$. Similarly we abbreviate the prediction $M_t = M_t(x_t)$ and $M_{1:t} = \prod_{s=1}^t M_s$. We usually reserve $w, w_t \in \mathcal{W}$ to denote the weights over experts used by an algorithm and $a \in \mathcal{W}$ for a fixed competitor. Given $a \in \mathcal{W}$, we let $A_t = \sum_{i=1}^n a^i p_t^i$ be the prediction of the mixture of experts over a and $A_{1:t} = \prod_{s=1}^t A_s$. The indicator function is denoted by $[\![test]\!] \in \{0, 1\}$ and equals 1 if the boolean *test* is true.

2. The Soft-Bayes algorithm

As discussed in the introduction, the Prod algorithm was originally designed for prediction with expert advice when competing with the best expert in hindsight. Let $w_1 \in \mathcal{W}$ be the initial (prior) weights, which we always take to be uniform $w_1 = \frac{1}{N}$ unless otherwise stated. Then in each round t the Prod algorithm predicts using a mixture over the experts (5) and updates its weights using a multiplicative update rule (6):

$$M_t = \sum_{i=1}^{N} w_t^i p_t^i \,. \qquad (5) \qquad\qquad w_{t+1}^i = \frac{w_t^i(1-\bar{\eta}\ell_t^i)}{\sum_{j=1}^N w_t^j(1-\bar{\eta}\ell_t^j)}\,, \qquad (6)$$

where $\ell_t^i = \ell_t(e_i)$ is the loss suffered by expert i in round t and $\bar{\eta} \in (0,1)$ is the learning rate. The algorithm only makes sense if $\ell_t^i \leq 1$, which is usually assumed. Recall our loss function is $\ell_t : \mathcal{W} \to \mathbb{R}$ is given by $\ell_t(w) = -\log M_t = -\log \sum_{i=1}^N w_t^i p_t^i$, which is convex but arbitrarily large. The key idea is to predict using Eq. (5), but replace the loss with the linearised loss, $\nabla \ell_t(w_t)^\top w$. A simple calculation shows that $\nabla \ell_t(w_t)_i = -p_t^i/M_t \leq 0$. Therefore the linearised losses are semi-bounded. If we instantiate Prod, but replace the loss of each expert with the linearised loss, then the resulting algorithm predicts like Eq. (5) and updates its weights by

$$w_{t+1}^i = \frac{w_t^i\left(1+\bar{\eta}\frac{p_t^i}{M_t}\right)}{\sum_{j=1}^N w_t^j\left(1+\bar{\eta}\frac{p_t^j}{M_t}\right)} = \frac{w_t^i\left(1+\bar{\eta}\frac{p_t^i}{M_t}\right)}{1+\bar{\eta}} = w_t^i\left(1-\eta+\eta\frac{p_t^i}{M_t}\right)\,, \qquad (7)$$

where $\eta = \bar{\eta}/(1+\bar{\eta})$ so that $\bar{\eta} = \eta/(1-\eta)$ and the second equality follows from the definition of $M_t = \sum_j w_t^j p_t^j$ and the fact that $\sum_i w_t^i = 1$. Notice that the computations of the prediction Eq. (5) and weight update Eq. (7) are both linear in the number of experts N. For reference, the EG and OGD algorithms also predicts like Eq. (5), but update their weights by

$$\text{EG:} \quad w_{t+1}^i = \frac{w_t^i \exp\left(\eta\frac{p_t^i}{M_t}\right)}{\sum_{j=1}^N w_t^j \exp\left(\eta\frac{p_t^j}{M_t}\right)} \qquad\qquad \text{OGD:} \quad w_{t+1}^i = \Pi\left(w_t + \eta\frac{p_t}{M_t}\right)_i\,, \qquad (8)$$

where Π is the projection onto the simplex with respect to the Euclidean norm. A careful examination of the EG update leads to a worrying observation: If w_t^i is close to zero and p_t^i is close to one, then p_t^i/M_t can be extremely large, causing w_{t+1}^i to be close to one and w_{t+1}^j to drop to nearly zero for all $j \neq i$. This makes EG unstable when the gradients are large. The OGD update can be even worse because the projection has the potential to concentrate the weights on a Dirac after which the regret can be infinite! In contrast, Prod behaves more conservatively since[1]

$$w_{t+1}^i = w_t^i\left(1-\eta+\eta\frac{p_t^i}{M_t}\right) = (1-\eta)w_t^i + \eta\frac{w_t^i p_t^i}{M_t} \leq (1-\eta)w_t^i + \eta\,,$$

[1] The second equality suggests that Prod and Soft-Bayes are closely related to exponential smoothing for probability estimation (Mattern, 2016).

where the inequality follows from the dominance property that $M_t \geq w_t^i p_t^i$ for all i and t. This means that even in the most extreme scenarios, the weight increases by at most η, which is usually tuned to be approximately $T^{-1/2}$.

Bayesian interpretation We now give two Bayesian interpretations of this algorithm. The first is to note that if $\eta = 1$, then

$$w_{t+1}^i = \frac{w_t^i p_t^i}{M_t} \qquad \text{and} \qquad M_{1:T} = \sum_{i=1}^N w_1^i p_{1:T}^i\,.$$

In this case w_t^i is the posterior of the Bayesian mixture over sources $(p^i)_i$ with prior w_1. While its regret relative to a single expert is at most $\mathcal{R}_T(e_i) \leq \log N$, the algorithm does not compete with convex combinations of experts. From a Bayesian perspective, there is no reason to believe that it should because the convex combinations lies outside the class of the learner. On the other hand, if the learning η is chosen to be close to zero, then the Prod update has the effect of 'slowing down Bayes' to ensure it does not concentrate too fast on a single promising expert. The multiplicative/additive nature of the update also means that experts can make big mistakes while losing at most $(1-\eta)$ of their current weight. In contrast, a 'slow' update derived from the exponential weights algorithm that looks like $w_{t+1}^i \propto w_t^i \exp(\eta \log(p_t^i/M_t))$ still reduces the weight of an expert to zero if $p_t^i = 0$.

The second interpretation comes from the sleeping expert framework (Freund et al., 1997) that allows experts to 'fall asleep' and abstain from predicting in some rounds. If the weights are normalised appropriately, then this is equivalent to assuming the sleeping experts defer their vote to the wakeful, which for them is equivalent to predicting like the mixture M_t (Chernov and Vovk, 2009). We consider a smooth version of this idea, where an expert can be 'sleepy' and predict partially like the mixture. From a given expert p^i, we build the meta-expert $\tilde{p}^i$ such that for all time steps t:

$$\tilde{p}_t^i := (1-\eta)M_t + \eta p_t^i \qquad \text{and thus} \qquad \tilde{p}_{1:T}^i = \prod_{t=1}^T ((1-\eta)M_t + \eta p_t^i)\,,$$

where $\eta \in (0,1]$ and M_t is now defined as a mixture of the meta-experts $\tilde{p}^i$:

$$M_{1:T} = \sum_{i=1}^N w_1^i \tilde{p}_{1:T}^i \tag{9}$$

Note that since both p and M are predictors the convex combination of these is also a predictor. Such self-referential constructions have been noted in the past, for example by Koolen et al. (2012). The main point is that the meta-experts are not normal predictors because they depend on the learner M. Nevertheless, they are useful for analysis and intuition. With this view of M we note that all the usual properties of Bayesian predictors hold. In particular:

- The posterior weight of the ith meta-expert is

$$w_{t+1}^i = w_1^i \frac{\tilde{p}_{1:t}^i}{M_{1:t}} = w_t^i \frac{\tilde{p}_t^i}{M_t} = w_t^i \left(1 - \eta + \eta \frac{p_t^i}{M_t}\right)\,.$$

- The Bayes prediction over the class of meta-experts is $M_t = \displaystyle\sum_{i=1}^{N} w_t^i \tilde{p}_t^i$.
- The weights are properly normalised: $\displaystyle\sum_{i=1}^{N} w_t^i = 1$ for all $t \in [T]$.

Based on these observations we call the algorithm defined by the prediction in Eq. (5) and updates in Eq. (7), or equivalently by Eq. (9), the *Soft-Bayes* algorithm.

Regret relative to a single expert We start the theoretical results with a simple bound on the regret relative to a single expert.

Theorem 1 *For the Soft-Bayes algorithm, $\mathcal{R}_T(e_i) = \ln \frac{p^i_{1:T}}{M_{1:T}} \leq \frac{1}{\eta} \ln \frac{1}{w_1^i}$ for all i.*

Proof Using dominance and the definition of concavity applied to the function $\ln(\cdot)$:

$$\begin{aligned}\forall i \in [N] : \ln M_{1:T} &\geq \ln w_1^i \tilde{p}^i_{1:T} = \ln w_1^i + \sum_t \ln((1-\eta)M_t + \eta p_t^i) \\ &\geq \ln w_1^i + (1-\eta)\sum_t \ln M_t + \eta \sum_{t=1}^{T} \ln p_t^i = \ln w_1^i + (1-\eta)\ln M_{1:T} + \eta \ln p^i_{1:T} \,.\end{aligned}$$

Therefore $\eta \ln M_{1:T} \geq \eta \ln p^i_{1:T} + \ln w_1^i$, and the proof is completed by rearrangement. ■

If the goal is to compete with the best expert only, then setting $\eta = 1$ is optimal, which makes Prod equivalent to the standard Bayesian algorithm over (e_i). As an aside, in Appendix F we present a simple setup where we recover several well-known algorithms by using Soft-Bayes with specific simple learning rates.

3. Failure of EG and OGD

As remarked in the introduction, the EG and OGD algorithms can become unstable when the gradients are uncontrolled. Here we demonstrate this with a carefully crafted example that best illustrates the issue.

Theorem 2 *If $N = 2$ and $w_1 = (1/2, 1/2) \in \mathcal{W}$ is the uniform prior, then for any learning rate $\eta > 0$ there exists a sequence of predictors such that the regret of EG and OGD is $\Omega(T^{1-\varepsilon})$ for all $\varepsilon \in (0,1)$.*

The proof is given in Appendix D and depends on a simple example where the experts predict Dirac measures with disjoint support (they always disagree). In the first $T/2$ rounds the first expert is always wrong and for the next $T/2$ rounds the correctness of the experts alternates. If the learning rate is sufficiently large, then the weights of the EG algorithm oscillate wildly, which leads to a *super-exponential* regret. The only way to avoid this calamity is to choose a learning rate so small that EG barely learns at all, in which case it suffers near-linear regret. For OGD the regret can even be infinite in this example. A naive attempt to fix the EG algorithm is to replace the experts with 'meta-experts' $p^{\delta i}$ defined by $p_t^{\delta i} := (1-\delta)p_t^i + \delta/|\mathcal{X}|$. This ensures that $M_t = \sum_i w_t^i p_t^{\delta i} \geq \delta/|\mathcal{X}| \approx 1/c$ for all t. While this does prevent super-exponential regret, it does not solve the problem. Compared to a mixture of the base experts p^i, the mixture of the meta-experts $p^{\delta i}$ can suffer a regret of $T\delta$. Hence, considering the bound in Eq. (2), the optimal balance is for $\delta \approx T^{-1/4}$ leading to a bound of $O(T^{3/4})$, which is much worse than the $O(T^{1/2})$ regret that we prove for Prod. A similar correction is possible for OGD, but does not seem worthwhile in light of the above discussion.

4. Regret against a mixture

Recall that for any $a \in \mathcal{W}$, the mixture predictor is $A_t = \sum_{i=1}^N a^i p_t^i$ and $A_{1:T} = \prod_{t=1}^T A_t$. Let $\mathcal{M}^* := \{i \mid \exists t \in [T] : p_t^i = \max_j p_t^j\}$ be the set of experts that predict at least as well as any other expert at least once, and let $m = |\mathcal{M}^*| \leq N$.

Theorem 3 *For any $\eta \in (0,1)$, the regret of the Soft-Bayes algorithm M is bounded by:*

$$\mathcal{R}_T(a) = \ln \frac{A_{1:T}}{M_{1:T}} \leq \frac{1}{\bar\eta} \ln N + \bar\eta m T + m \ln \frac{N}{m} + \ln N\,, \qquad \textit{where } \bar\eta := \frac{\eta}{1-\eta}\,.$$

The learning rate $\bar\eta$ is optimised by $\bar\eta = \sqrt{\frac{\ln N}{Tm}}$ and for this choice

$$\begin{aligned} \mathcal{R}_T(a) = \ln \frac{A_{1:T}}{M_{1:T}} &\leq 2\sqrt{Tm \ln N} + m \ln \frac{N}{m} + \ln N \\ \text{and } \mathcal{R}_T(a) \qquad &\leq 2\sqrt{TN \ln N} + \ln N\,. \end{aligned}$$

To prove this theorem we need the following lemma (proof in Appendix B).

Lemma 4 *Let $\eta \in (0,1), a \in \mathcal{W}$, and $q \in [0,\infty)^N$,*

$$\ln \sum_i a_i q_i \leq \frac{1}{\eta} \sum_i a_i \ln\left(1 - \eta + \eta q_i\right) + \max_i \ln\left(1 + \frac{\eta}{1-\eta} q_i\right).$$

Proof (Theorem 3) By Lemma 4 and using $w_{t+1}^i = w_t^i(1-\eta)\left(1 + \frac{\eta}{1-\eta}\frac{p_t^i}{M_t}\right)$, for any $t \in [T]$:

$$\begin{aligned} \ln \frac{A_t}{M_t} = \ln \sum_{i=1}^N a^i \frac{p_t^i}{M_t} &\leq \frac{1}{\eta} \sum_i a^i \ln\left(1 - \eta + \eta \frac{p_t^i}{M_t}\right) + \max_{i \leq N} \ln\left(1 + \frac{\eta}{1-\eta}\frac{p_t^i}{M_t}\right) \\ &= \frac{1}{\eta} \sum_i a^i \ln \frac{w_{t+1}^i}{w_t^i} + \max_i \ln\left(\frac{w_{t+1}^i}{w_t^i}\right) - \ln(1-\eta)\,. \end{aligned}$$

The first term telescopes when summed over time, but more effort is required to control the second since the index i of $\max_i$ can change with time. The idea is to introduce all the missing $\ln w_{t+1}^i / w_t^i$ terms for all i that can be the $\max_i$ at some step t, that is all $i \in \mathcal{M}^*$. This comes at a cost of $\ln(1-\eta)$ for each of them since $w_{t+1}^i / w_t^i \geq 1 - \eta$:

$$\ln \frac{A_t}{M_t} \leq \frac{1}{\eta} \sum_i a^i \ln \frac{w_{t+1}^i}{w_t^i} - m \ln(1-\eta) + \sum_{i \in \mathcal{M}^*} \ln \frac{w_{t+1}^i}{w_t^i}\,. \tag{10}$$

We can now telescope the series over time, also using $-\ln(1-\eta) \leq \frac{\eta}{1-\eta} = \bar{\eta}$ (Lemma 13),

$$\ln \frac{A_{1:T}}{M_{1:T}} = \sum_{t=1}^{T} \ln \frac{A_t}{M_t} \leq \frac{1}{\eta} \sum_i a^i \ln \frac{w^i_{T+1}}{w^i_1} + \bar{\eta} m T + \sum_{i \in \mathcal{M}^*} \ln \frac{w^i_{T+1}}{w^i_1} \,.$$

Using $w^i_1 = 1/N$ it holds that $\sum_i a^i \ln(w^i_{T+1}/w^i_1)$ is maximised when $w^i_{T+1} = a^i$ and similarly $\sum_{i \in \mathcal{M}^*} \ln(w^i_{T+1}/w^i_1)$ is maximised when $w^i_{T+1} = 1/m$. Therefore

$$\begin{aligned} \ln \frac{A_{1:T}}{M_{1:T}} &\leq \frac{1}{\eta} \mathrm{RE}(a \| w_1) + \bar{\eta} m T + m \ln \frac{N}{m} \quad \leq \frac{1}{\eta} \ln N + \bar{\eta} m T + m \ln \frac{N}{m} \qquad (11) \\ &= \frac{1}{\bar{\eta}} \ln N + \bar{\eta} m T + m \ln \frac{N}{m} + \ln N \,. \end{aligned}$$

If we replace $\mathcal{M}^*$ with all the N experts in Eq. (10), we obtain $m = N$ giving the second bound. ■

5. Self-confident bounds

Self-confident bounds were introduced by Auer and Gentile (2000) in online prediction to build algorithms that can perform better when the sequence is easy, instead of considering that all sequences are worst cases. These bounds depend on the loss of the competitor. For Prod, Gaillard et al. (2014) derived second order self-confident bounds that depend on the excess losses, that is, the difference between the instantaneous loss of the learner and that of one of the experts. It is unclear how to use these bounds with the gradient trick, but we provide bounds of a very similar flavour.

Theorem 5 *Consider the Soft-Bayes algorithm with learning rate $\eta \in (0, 1)$ and $\bar{\eta} = \frac{\eta}{1-\eta}$. Then*

$$\begin{aligned} \mathcal{R}_T(a) = \ln \frac{A_{1:T}}{M_{1:T}} &\leq \frac{1}{\bar{\eta}} \ln N + \bar{\eta} \max_{i \leq N} \sum_{t=1}^{T} \left(\frac{p^i_t}{M_t} - 1 \right)^2 + \ln N \qquad (12) \\ &\leq \frac{1}{\bar{\eta}} \ln N + \bar{\eta} T \max_{i \leq N, t \leq T} \left(\frac{p^i_t}{M_t} - 1 \right)^2 + \ln N \,. \qquad (13) \end{aligned}$$

The proof is in Appendix C. Let $C_2 := \max_{i,t} \left(\frac{p^i_t}{M_t} - 1 \right)^2$, then Equation (13) is optimized for $\bar{\eta} := \sqrt{\frac{\ln N}{T C_2}}$ leading to a regret bound of $2\sqrt{T C_2 \ln p} + \ln N = \max_{i,t} 2 \left(\frac{p^i_t}{M_t} - 1 \right) \sqrt{T \ln N} + \ln N$. Furthermore, since the learning rate is monotonically decreasing, this bound is suitable for an online learning rate. The case for Eq. (12) is more complicated: as noted by Cesa-Bianchi et al. (2007); Gaillard et al. (2014) for Prod and others elsewhere this sort of bound depends on the best expert in hindsight and thus does not lead to a monotone decreasing learning rate in general. Gaillard et al. (2014) circumvent this issue by using one learning rate per expert, at the cost of only a multiplicative $O(\ln \ln T)$ factor in the loss. We perform a similar transformation in Appendix E without this additional factor. As discussed in

Section 3, since p_t^i/M_t can be large, the quadratic dependence on $\max_{i,t}(p_t^i/M_t - 1)^2$ can be poor. For this reason we show that with a small additional cost the quadratic term can be replaced with a linear term.

Theorem 6 *Consider the Soft-Bayes algorithm with learning rate* $\eta \in (0, 1)$ *and let* $C_1 = \sum_{t=1}^T \max_{i \le N} \left(\frac{p_t^i}{M_t} - 1\right)$*. Then*

$$\mathcal{R}_T(a) = \ln \frac{A_{1:T}}{M_{1:T}} \le \min \left\{ C_1, \quad \frac{1}{\eta} \ln N + \frac{\eta}{2} C_1 + \eta^2 T \right\} .$$

The proof is in Appendix C. If the learning rate is chosen to be $\eta = \sqrt{\frac{2 \log N}{C_1}}$ then the theorem shows that

$$\mathcal{R}_T(a) \le \min \left\{ C_1, \sqrt{2 C_1 \ln N} + \frac{2T \ln N}{C_1} \right\} .$$

Observe that if a single expert i is always the best predictor, M_t becomes close to p_t^i and $\frac{p_t^i}{M_t} - 1$ becomes close to 0 and then $C_1 << T$; However for this case learning will likely still be slower than with second-order self-confident bounds.

In another case where for example $A_t = \sum_{i \in \mathcal{M}^*} \frac{1}{m} p_t^i$, that is, at worst the best expert alternates uniformly between a subset $\mathcal{M}^*$ of $m = |\mathcal{M}^*|$ experts, M_t cannot become close to all the p_t^i at the same time and then $C_1 = O(T)$, which still makes $\mathcal{R}_T(a) \le O(\sqrt{T})$ (omitting other dependencies); Furthermore, since this means that M learns, M_t should become close to $\sum_{i \in \mathcal{M}^*} \frac{1}{m} p_t^i$ and thus $C_1 \approx mT$ so we should have $\mathcal{R}_T(a) \le O(\sqrt{mT \ln N})$, hence possibly providing good guarantees against the best subset of the experts. By contrast, a second-order self-confident bounds would only provide a guarantee of $O(m\sqrt{T \ln N})$.

Moreover, upper bounding C_1 with $T \max_{i,t} \frac{p_t^i}{M_t}$ in Theorem 6 and then optimizing η as above gives the result of Eq. (4) for $C = \max_{i,t} \frac{p_t^i}{M_t}$.

Also note that C_1 is monotonically increasing with T and thus the learning rate can be updated online.

6. Online bounds

We now provide a fully 'online' algorithm that does not require advance knowledge of the time horizon T or the number of 'sometimes optimal experts' m in advance. Surprisingly, we could not simply replace the learning rating η with a time-varying version $\eta_t \approx 1/\sqrt{t}$ and adapt the proofs in a straightforward manner. The reason seems to be that with a fixed rate of $\eta := 1/\sqrt{T}$, the weights can decay as $w_1^i(1 - 1/\sqrt{T})^t \approx w_1^i \exp(-t/\sqrt{T})$ after t steps, whereas for a time-varying learning rate $\eta_t := 1/\sqrt{t}$, the weights can decay as fast as $w_1^i \prod_{s=1}^t (1 - 1/\sqrt{t}) \approx w_1^i \exp(-\sqrt{t})$ (consider for example $t = \sqrt{T}$). An easy solution is to use the doubling trick (Cesa-Bianchi et al., 1997), which was the approach taken in the analysis of the original Prod algorithm (Cesa-Bianchi et al., 2007). Another option is to use an exponential rescaling with renormalization, which was used for ML-Prod (Gaillard et al., 2014). We show instead a different online correction of the update rule based on a special form of the fixed-share rule (Herbster and Warmuth, 1998). When $0 < \eta_{t+1} \leq \eta_t \leq 1$, we use the following online correction term applied to the update rule:[2]

$$w^i_{t+1} := \underbrace{w^i_t \left(1 - \eta_t + \eta_t \frac{p^i_t}{M_t}\right) \frac{\eta_{t+1}}{\eta_t}}_{\text{update}} + \underbrace{\left(1 - \frac{\eta_{t+1}}{\eta_t}\right) w^i_1}_{\text{online correction}} . \tag{14}$$

Lemma 7 *The online update rule of Eq. (14) has the following properties:*

$$\begin{aligned}
w_{t+1} &\in \mathcal{W} && \textit{(normalized)} && (15)\\
w^i_t &\geq w^i_1 \left(1 - \frac{\eta_t}{\eta_{t-1}}\right) && \textit{(restarting)} && (16)\\
\ln\left(1 - \eta_t + \eta_t \frac{p^i_t}{M_t}\right) &\leq \ln \frac{w^i_{t+1}}{w^i_t} + \ln \frac{\eta_t}{\eta_{t+1}} && \textit{(telescoping)} && (17)\\
0 &\leq \ln \frac{w^i_{t+1}}{w^i_t} + \ln \frac{\eta_t}{\eta_{t+1}} + \frac{\eta_t}{1 - \eta_t} && \textit{(loss injection)} && (18)\\
\frac{1}{\eta_t} \ln\left(1 - \eta_t + \eta_t \frac{p^i_t}{M_t}\right) &\leq \frac{1}{\eta_{t+1}} \ln \frac{w^i_{t+1}}{w^i_1} - \frac{1}{\eta_t} \ln \frac{w^i_t}{w^i_1} && \textit{(1/}\eta\textit{-telescoping)} && (19)
\end{aligned}$$

The restarting property ensures that the weights are never too small, enabling the mixture to 'restart' the learning process and offer tracking guarantees. The loss injection property will be useful in the theorems to force some series to telescope by 'injecting' some additional loss as was done offline in the proof of Theorem 3.

Proof Equation (15) follows from the fact that the Soft-Bayes update rule keeps the weights normalized, and so does the fixed-share rule. Starting from Eq. (14), Eq. (16) follows from dropping the l.h.s. of the +. Equation (17) follows from dropping the r.h.s. of the +, dividing by w^i_t, taking the log then rearranging. Equation (18) follows from Eq. (17) by taking $p^i_t = 0$ and from $-\ln(1 - x) \leq \frac{x}{1-x}$ (Lemma 13) and rearranging. For Eq. (19), dividing by w^i_1, taking $\beta = \frac{\eta_{t+1}}{\eta_t}$ we have:

$$\begin{aligned}
\ln \frac{w^i_{t+1}}{w^i_1} &= \ln \left(\beta \frac{w^i_t}{w^i_1}\left(1 - \eta_t + \eta_t \frac{p^i_t}{M_t}\right) + (1 - \beta)\right)\\
&\geq \beta \ln \frac{w^i_t}{w^i_1}\left(1 - \eta_t + \eta_t \frac{p^i_t}{M_t}\right) = \frac{\eta_{t+1}}{\eta_t} \ln \frac{w^i_t}{w^i_1}\left(1 - \eta_t + \eta_t \frac{p^i_t}{M_t}\right) ,
\end{aligned}$$

where we used Jensen's inequality, with $\beta \in [0, 1]$ since $\eta_{t+1} \geq \eta_t$ as required in Eq. (14). Dividing by η_{t+1} and rearranging gives the result. ■

We will not provide formal results for the self-confident learner but we make the following observation.

[2] Note that this online correction can also be used with EG.

Remark 8 *Using the online correction rule of Eq. (14) with the self-confident learning rate of Theorem 6 gives*

$$\frac{\eta_{t+1}}{\eta_t} = \sqrt{\frac{\sum_{k=1}^{t-1}(\max_i \frac{p_k^i}{M_k} - 1)}{\sum_{k=1}^{t}(\max_i \frac{p_k^i}{M_k} - 1)}} = \sqrt{1 - \frac{\max_i \frac{p_t^i}{M_t} - 1}{\sum_{k=1}^{t}(\max_i \frac{p_k^i}{M_k} - 1)}} \approx 1 - \frac{\max_i \frac{p_t^i}{M_t} - 1}{2\sum_{k=1}^{t}(\max_i \frac{p_k^i}{M_k} - 1)}\,.$$

Let $i_t := \arg\max_i p_t^i$. On a step t where the mixture makes a bad prediction, $\frac{p_t^{i_t}}{M_t} \leq 1/w_t^{i_t}$ is large so the weight $w_t^{i_t}$ is small. Considering the restarting property of Eq. (16), this means that the weight $w_t^{i_t}$ (in particular) receives a boost $1 - \frac{\eta_{t+1}}{\eta_t}$ toward its prior, hence helping the mixture coping with experts that suddenly become good predictors after a long period of bad predictions—except that this may be one time step too late. Indeed, for the self-confident learner the ratio $\frac{\eta_{t+1}}{\eta_t}$ can be close to 1 when the mixture predicts well, which means that the weights of bad predictors may still decrease exponentially fast—potentially resulting in large instantaneous losses if they become good predictors later. To prevent this, we advise replacing $\frac{\eta_{t+1}}{\eta_t}$ with $\min\left\{\frac{\eta_{t+1}}{\eta_t}, \sqrt{\frac{t}{t+1}}\right\}$ in Eq. (14), which ensures that the weights do not decrease faster than $O(1/t)$, while still retaining the quicker restarting property of the self-confident learning rate.

The following generic bound will be used for the various proofs.

Lemma 9 *For any sequence of monotone decreasing learning rates $\eta_t \in (0, 1)$, when using the update rule of Eq. (14), the regret of the mixture M of the N experts with prior weights w_1 compared to the best fixed combination A with weights a is bounded by:*

$$\ln\frac{A_{1:T}}{M_{1:T}} \leq \frac{1}{\eta_{T+1}}\mathrm{RE}(a\|w_1) + \ln\frac{\eta_1}{\eta_{T+1}} + \sum_{t=1}^{T}\frac{\eta_t}{1-\eta_t} + \sum_{t=1}^{T}\max_{i\leq N}\ln\frac{w_{t+1}^i}{w_t^i}.$$

Proof Starting from Lemma 4 and similarly to the proof of Theorem 3 we have:

$$\begin{aligned}\ln\frac{A_t}{M_t} &\leq \sum_{i=1}^{N} a^i\frac{1}{\eta_t}\ln\left(1-\eta_t+\eta_t\frac{p_t^i}{M_t}\right) + \max_i \ln\left(1+\frac{\eta_t}{1-\eta_t}\frac{p_t^i}{M_t}\right)\\ &= \sum_{i=1}^{N} a^i\frac{1}{\eta_t}\ln\left(1-\eta_t+\eta_t\frac{p_t^i}{M_t}\right) + \max_i \ln\left(1-\eta_t+\eta_t\frac{p_t^i}{M_t}\right) - \ln(1-\eta_t)\,. \qquad (20)\end{aligned}$$

Using Lemma 13, $-\ln(1-\eta_t) \leq \frac{\eta_t}{1-\eta_t}$ along with the $1/\eta$-telescoping property of Eq. (19) on the term in the sum and the telescoping property of Eq. (17) on the term in the max gives:

$$\ln\frac{A_t}{M_t} \leq \sum_i a^i\left(\frac{1}{\eta_{t+1}}\ln\frac{w_{t+1}^i}{w_1^i} - \frac{1}{\eta_t}\ln\frac{w_t^i}{w_1^i}\right) + \max_i\ln\frac{w_{t+1}^i}{w_t^i} + \ln\frac{\eta_t}{\eta_{t+1}} + \frac{\eta_t}{1-\eta_t}.$$

Summing $\ln\frac{A_t}{M_t}$ over t and telescoping the first term $\sum_i a^i(\cdot)$ and the third term (but not the second one) leads to:

$$\ln\frac{A_{1:T}}{M_{1:T}} \leq \sum_{i=1}^{N} a^i\ln\frac{w_{T+1}^i}{w_1^i} + \sum_{t=1}^{T}\max_i\ln\frac{w_{t+1}^i}{w_t^i} + \ln\frac{\eta_1}{\eta_{T+1}} + \sum_{t=1}^{T}\frac{\eta_t}{1-\eta_t}.$$

Finally, taking the worst case $w^i_{T+1} = a^i$ and rearranging gives the result. ■

We are now ready to show the online bounds. First we track only the time step t.

Theorem 10 *When using the update rule in Eq. (14) with the learning rate* $\eta_t := \sqrt{\frac{\ln N}{2Nt}}$, *we have the following regret bound against the best fixed convex combination A of the experts:*

$$\ln \frac{A_{1:T}}{M_{1:T}} \leq 2\sqrt{2(T+1)N \ln N} + (\tfrac{1}{2}N + \ln N)\ln(T+1) + \ln N.$$

Proof We start from Lemma 9, and work on the $\sum_t \max_i$ term. As in the proof of Theorem 3, the main idea is to make this term telescope by injecting some additional positive terms. This is done by using the loss injection property of Eq. (18) repeatedly (starting at $t = 1$) on all the $N-1$ experts i that are not already in the sum, leading to:

$$\begin{aligned}\sum_{t=1}^T \ln \frac{A_t}{M_t} &\leq \frac{1}{\eta_{T+1}} \underbrace{\mathrm{RE}(a\|w_1)}_{\leq \ln N} + N \ln \frac{\eta_1}{\eta_{T+1}} + N \sum_{t=1}^T \underbrace{\frac{\eta_t}{1-\eta_t}}_{=\eta_t + \frac{\eta_t^2}{1-\eta_t}} + \underbrace{\sum_i \ln \frac{w^i_{T+1}}{w^i_1}}_{\leq 0} \\ &\leq \frac{1}{\eta_{T+1}} \ln N + N \ln \frac{\eta_1}{\eta_{T+1}} + N \sum_{t=1}^T \eta_t + N \sum_{t=1}^T \frac{\eta_t^2}{1-\eta_t}.\end{aligned}$$

Taking $\eta_t := \sqrt{\frac{\ln N}{2Nt}}$, and since $1 - \eta_t \geq \frac{1}{2}$, we obtain:

$$\begin{aligned}\sum_{t=1}^T \ln \frac{A_t}{M_t} &\leq \sqrt{2(T+1)N \ln N} + N \ln \sqrt{T+1} + \sqrt{\frac{N \ln N}{2}} \underbrace{\sum_{t=1}^T \frac{1}{\sqrt{t}}}_{\leq 2\sqrt{T+1}} + \ln N \underbrace{\sum_{t=1}^T \frac{1}{t}}_{\leq 1 + \ln T} \\ &\leq 2\sqrt{2(T+1)N \ln N} + (\tfrac{1}{2}N + \ln N)\ln(T+1) + \ln N.\end{aligned}$$

■

This regret is only a factor $\sqrt{2} \approx 1.41$ worse on the leading term than the corresponding offline bound using $\eta = \sqrt{\ln(N)/(NT)}$, which is better than the $\sqrt{2}/(\sqrt{2}-1) \approx 3.41$ factor that would be obtained via the doubling trick.

6.1. Sparse expert set: Tracking $\mathcal{M}^*_t$

We would like to have an online bound of the order $O(\sqrt{Tm \ln N})$ as in Theorem 3 where m is the number of 'good' experts. Interestingly, setting naively $\eta_t = \sqrt{\ln N/(2tm_t)}$ works, but not for the naive reasons. Indeed, merely adapting the proof of Theorem 3 leads to $\sum_{i \in \mathcal{M}^*} \sum_t \eta_t \approx m\sqrt{T \ln N}$ regret in the worst case where only one expert is good for $T - m + 1$ steps ($m_t = 1$), and on the $m - 1$ last steps the other experts are the best predictors. Let T_i the first time step at which expert i is the best expert,[3] that is $T_i :=$

[3] Breaking ties can be done most favourably by picking as the best expert one that was already counted as such, to avoid introducing new 'good' experts.

$\min\{t : p_t^i = \max_j p_t^j\}$. Let $\mathcal{M}_t^* := \{i : T_i < t\}$ be the set of experts that have been the best expert at least once (strictly) before time step t, let $m_t := \max\{1, |\mathcal{M}_t^*|\}$, $\mathcal{M}^* := \mathcal{M}_{T+1}^*$ and $m := m_{T+1}$.

Theorem 11 *When using the update rule in Eq. (14) with the learning rate $\eta_t := \sqrt{\frac{\ln N}{2m_t t}}$, we have the following regret bound against the best fixed convex combination A of the experts:*

$$\ln \frac{A_{1:T}}{M_{1:T}} \leq 2\sqrt{2m(T+1)\ln N} + (m + \ln N)\ln T + m \ln \frac{N}{m}$$
$$+ 1.2m + \sqrt{\tfrac{1}{2}\ln N}(1 + \ln m) + 3.5 \ln N.$$

The proof is in Appendix C. Again, we only get a $\sqrt{2}$ factor on the leading term compared to the offline version. One drawback of this algorithm is that any expert that is the best one even only once will be counted in m. It may be desirable to forget about experts that have not been best for a long time and thus decrease m. This is left as an open problem.

6.2. Shifting regret

In this subsection we show that the online version of Prod can compete with the best sequence of convex combinations of experts. Let $1 \leq K \leq T$ and A be a sequence of constant competitors $A^1, A^2 \dots A^K$. Each competitor A^k starts at step T_k and ends at step $T_{k'} = T_{k+1} - 1$. Thus, assuming $T_{K'} = T$, we have $A_{1:T} = A^1_{1:T_{1'}} A^2_{T_2:T_2} \dots A^K_{T_K:T}$. Each competitor A^k has associated weights a_k^i that remain fixed on the interval $T_k : T_{k'}$.

With the learning rate of Theorem 10, we readily obtain a shifting regret bounded in $O(K \ln(T)\sqrt{TN \ln N})$, but by tuning the learning rate and still without prior knowledge of K it can be reduced to $O((K + \ln T)\sqrt{TN \ln N})$:

Theorem 12 *When using the learning rate $\eta_t = \sqrt{\frac{\ln N}{2Nt}} \ln(t+3)$ with the update rule in Eq. (14), for $T \geq 2$ the K-shifting regret is bounded by:*

$$\ln \frac{A_{1:T}}{M_{1:T}} \leq \sqrt{2(T+1)N \ln N}\left(\ln(T+3) + K\left(\frac{2}{\ln N} + \frac{1}{\ln T}\right)\right)$$
$$+ \frac{5}{4}\frac{\ln N}{N}(1 + \ln T)^3 + \frac{N}{2}\ln(T+1)\,.$$

If K were known in advance, then the learning rate can be tuned so that the regret is $O(\sqrt{TKN \ln(NT)})$. In the absence of this knowledge one can still have the K inside the square root by competing with several learning rates as discussed in the conclusion. The proof is in Appendix C.

7. Conclusion

We have shown new regret guarantees for the Prod algorithm when competing against a convex combination of the experts. In particular, we proved that $\mathcal{R}_T = O(\sqrt{TN \ln N})$, which unlike EG does not depend on the (possibly unbounded) largest gradient. The online version of the algorithm uses a special form of the fixed-share rule, which simultaneously makes the algorithm truly online, computable in $O(N)$ steps per round and also enjoys strong shifting regret guarantees. A short discussion and some open questions follow.

Alternative approaches As discussed in Section 3, the EG algorithm fails when it encounters large gradients. Since these only occur when the weights are close to zero, one might try an approach based on follow-the-regularised-leader or mirror descent (Hazan, 2016, for an overview). The natural choice of regulariser is $R(w) = -\sum_i \log w^i$, which after a long calculation can be shown to eliminate the dependence on the largest gradient. The regret guarantee is slightly worse than for Prod, as it has a logarithmic dependence on T rather than N so that $\mathcal{R}_T = O(\sqrt{NT \log(T/N)})$. Furthermore, the algorithm does not immediately have tracking guarantees and the projection step involves a line-search, which naively requires a computation time of $O(N \log(T))$ per round.

Mixtures of learning rates Many of the results in the previous sections have depended on a specific 'optimal' choice of learning rate that allows Prod or its online variant to adapt to specific kinds of structure in the data. The downside of fixing a single learning rate is that if the structure of interest is not present, then the algorithm may perform badly. In many cases it is possible to derive a clever scheme for adapting the learning rate online to achieve the best of several worlds. An alternative is to exploit the special property of the log-loss and to simply create a meta-agent that mixes over a discrete set of predictors. For example, in the offline case one can predict using the Bayesian mixture over a set of Soft-Bayes predictors with learning rates $(\eta_i)_{i=0}^K$ where $\eta_i = 2^{-i}$ and $K = \log_2(T)$. If the uniform prior is used, then the Bayesian mixture will suffer an additional regret of only $\ln(K) = \ln \log_2(T)$ relative to the best soft-Bayes in the class. Provided that the optimal learning rate lies in $[1/T, 1]$, then this mixture will compete with a learning rate that is at most a factor of 2 off for which the penalty is just a factor of 2 at worst. The additional regret is small enough to be insignificant. More concerning is that the computation cost becomes $O(KN)$ per round. In practice, however, this procedure is easily parallelised and often leads to significant improvement. Additionally, at almost no additional computation cost we can build a switching mixture (Herbster and Warmuth, 1998; Veness et al., 2012a) between the individual experts (full Bayes, $\eta = 1$) and a Soft-Bayes mixture with rate $\sqrt{\ln N/(2Nt)}$, to enjoy logarithmic loss ($\ln(NT)$, the cost of switching) against segments of the sequence where a single expert is the best one, and revert to $\sqrt{T}$ loss for segments where we need to compete against a combination of the experts. The computation time is still in $O(N)$ per round.

Computationally efficient logarithmic regret The holy grail would be an $O(N)$-time algorithm with logarithmic regret and no dependence on the largest observed gradients of the (linearised) loss. A reasonable conjecture is that this is not possible, a proof of which would be quite remarkable. The most efficient algorithm with logarithmic regret is online Newton step for which the best known computation time is $O(N^2)$, which is prohibitively large for $n \gtrsim 10^4$. Furthermore, the online Newton step suffers from the same catastrophic failures as EG when poorly performing predictors suddenly become good. This limitation can be overcome via additional regularisation as for mirror descent above, but naively this pushes the computation cost to at least $O(N^4)$ because the projection step becomes more complex.

Different frameworks Another interesting direction is to extend the analysis beyond the log-loss and the linear mixing. Regarding losses, the most natural first step might be to examine the exp-concave case. Alternatively one could generalise the linear mixture to (say) a geometric mixture and see if Prod or similar can be applied to this practical setting (Mattern, 2016).

Acknowledgments

We would like to thank the following people for their help: Marc Bellemare, Guillaume Desjardins, Marc Lanctot, Andrew Lefranq, Jan Leike, Rémi Munos, Georg Ostrovski, Bernardo Avila Pires, David Saxton, Joel Veness.

Appendix A. Technical results

Lemma 13

$$\forall x < 1 : -\ln(1-x) \leq \frac{x}{1-x}.$$

Proof

$$-\ln(1-x) = \ln\left(\frac{1}{1-x}\right) = \ln\left(1 + \frac{x}{1-x}\right) \leq \frac{x}{1-x}$$

where the inequality follows from $\ln(1+x) \leq x$. ■

Lemma 14 (Love, 1980)

$$\forall x \geq 0 : \ln(1+x) \geq \left(\frac{1}{x} + \frac{1}{2}\right)^{-1} = \frac{x}{1+x/2} = \frac{2x}{2+x}.$$

Proof

$$\begin{aligned} \text{Let } f(x) &:= (2+x)\ln(1+x) - 2x \\ \text{then } f'(x) &= \ln(1+x) + \frac{2+x}{1+x} - 2 = \ln(1+x) + \frac{1}{1+x} - 1 \\ \text{and } f''(x) &= \frac{1}{1+x} - \frac{1}{(1+x)^2} = \frac{x}{(1+x)^2}. \end{aligned}$$

For all $x \geq 0$, since $f''(x) \geq 0$, f is convex, and since $f(0) = 0$ and $f'(0) = 0$ then $f(x) \geq 0$, which proves the result. ■

Lemma 15

$$\forall x \geq 0 : \quad \ln(1+x) \leq x - \frac{x^2/2}{1+x}.$$

Proof Let $f(x) := x - \frac{x^2/2}{1+x} - \ln(1+x)$. Then

$$f'(x) = 1 - \frac{x}{1+x} + \frac{x^2/2}{(1+x)^2} - \frac{1}{1+x} = \frac{x^2/2}{(1+x)^2}.$$

Since $f(0) = 0$, and $f'(x) > 0 \ \forall x > 0$, f is positive monotone increasing, which proves the result. ■

Corollary 16

$$\forall x \in (0, \tfrac{1}{2}] : \quad \frac{1}{x}\ln\frac{1}{1-x} - 1 \leq x/2 + x^2.$$

Proof Using Lemma 15 and $\frac{1}{1-x} = 1 + \frac{x}{1-x}$:

$$\begin{aligned}\ln \frac{1}{1-x} = \ln\left(1 + \frac{x}{1-x}\right) &\leq \frac{x}{1-x} - \frac{1}{2}\frac{\frac{x^2}{(1-x)^2}}{1+\frac{x}{1-x}} = \frac{x}{1-x} - \frac{1}{2}\frac{x^2}{1-x} \\ &= x + \frac{x^2}{1-x} - \frac{1}{2}\frac{x^2}{1-x} \\ &= x + \frac{x^2/2}{1-x}.\end{aligned}$$

Hence

$$\frac{1}{x}\ln\frac{1}{1-x} - 1 \leq \frac{x/2}{1-x} = x/2 + \frac{x^2/2}{1-x} \leq x/2 + x^2$$

where the last inequality holds if $x \leq \frac{1}{2}$. ■

Lemma 17

$$\forall x \geq 0, \forall \eta \in (0,1): \quad (x-1) \leq \frac{1}{\eta}\ln(1-\eta+\eta x) + \frac{\eta}{1-\eta}(x-1)^2.$$

Proof Using $\log(1+x) \geq \frac{x}{1+x}$:

$$\begin{aligned}\frac{1}{\eta}\ln(1-\eta+\eta x) = \frac{1}{\eta}\ln(1+\eta(x-1)) &\geq \frac{x-1}{1+\eta(x-1)} = (x-1) - \frac{\eta(x-1)^2}{1+\eta(x-1)} \\ &\geq (x-1) - \frac{\eta(x-1)^2}{1-\eta}\end{aligned}$$

where the last inequality holds with $x \geq 0$. Rearranging gives the result. ■

Lemma 18 *For all $t = 1, 2, 3, \ldots$:*

$$-\ln\left(1 - \frac{\ln(t+3)}{\ln(t+2)}\sqrt{\frac{t-1}{t}}\right) \leq \ln(t) + 1.6.$$

Proof By exhaustive search, the result holds for all $t \in [1..30]$. Now consider $t \geq 30$ for the rest of the proof. Observe that

$$\frac{\ln(t+3)}{\ln(t+2)} = 1 + \frac{\ln(1+1/(t+2))}{\ln(t+2)} \leq 1 + \frac{1}{(t+2)\ln(t+2)}.$$

Therefore:

$$\begin{aligned}-\ln\left(1 - \frac{\ln(t+3)}{\ln(t+2)}\sqrt{\frac{t-1}{t}}\right) &\leq \ln(t) - \ln\left(t - \left(1 + \frac{1}{(t+2)\ln(t+2)}\right)\sqrt{t(t-1)}\right) \\ &= \ln(t) - \ln\left(t(1 - \underbrace{\sqrt{1-1/t}}_{\leq 1-1/(2t)}) - \frac{\sqrt{t(t-1)}}{(t+2)\ln(t+2)}\right) \\ &\leq \ln(t) - \ln\left(\tfrac{1}{2} - \frac{1}{\ln(t+2)}\right) \quad \leq \ln(t) - \ln\left(\tfrac{1}{2} - \frac{1}{\ln(30+2)}\right) \quad \leq \ln(t) + 1.6\,.\end{aligned}$$

■

Appendix B. Reverse Jensens' inequalities

Lemma 19 *Let $\eta \in (0, \frac{1}{2}], a \in \mathcal{W}$, and $\forall i \in [N] : q_i \geq 0$ then:*

$$\ln \sum_{i=1}^{N} a_i q_i \leq \sum_{i=1}^{N} a_i \frac{1}{\eta} \ln \left(1 - \eta + \eta q_i\right) + \max_{i \leq N} \frac{\eta}{2} \left(q_i - 1\right) + \eta^2.$$

Proof Let $\ln_\eta(q) := \frac{1}{\eta} \ln(1 - \eta + \eta q)$, and let $\bar{\eta} := \frac{\eta}{1-\eta}$. By concavity of $\ln_\eta$, for $q \in [0, Q]$:

$$\begin{aligned} \ln_\eta(q) &\geq \ln_\eta(0) + \frac{q}{Q} \left(\ln_\eta(Q) - \ln_\eta(0)\right) \\ &= \frac{1}{\eta} \ln(1-\eta) + \frac{q}{Q} \left(\frac{1}{\eta} \ln \left[(1-\eta) \left(1 + \frac{\eta}{1-\eta} Q\right) \right] - \frac{1}{\eta} \ln(1-\eta) \right) \\ &= \frac{1}{\eta} \ln(1-\eta) + q \frac{\ln(1 + \bar{\eta} Q)}{\eta Q} \qquad =: g(q). \end{aligned}$$

Now since $\frac{\mathrm{d}}{\mathrm{d}q}(\ln q - g(q)) = \frac{1}{q} - \frac{\ln(1+\bar{\eta}Q)}{\eta Q}$, the maximum of $\ln q - g(q)$ is found at $\hat{q} = \frac{\eta Q}{\ln(1+\bar{\eta}Q)}$. Therefore:

$$\ln q - g(q) \leq \ln \hat{q} - g(\hat{q}) \leq \ln \frac{\eta Q}{\ln(1 + \bar{\eta} Q)} - \frac{1}{\eta} \ln(1-\eta) - 1. \tag{21}$$

Using Lemma 14, $\ln(1 + \bar{\eta} Q) \geq \frac{\bar{\eta} Q}{1 + \bar{\eta} Q / 2} = \frac{\eta Q}{1 - \eta + \eta Q/2}$ and thus:

$$\begin{aligned} \ln q - g(q) &\leq \ln(1 - \eta + \eta Q/2) - \frac{1}{\eta} \ln(1-\eta) - 1 \\ &= \ln(1 + \eta(Q/2 - 1)) - \frac{1}{\eta} \ln(1-\eta) - 1. \end{aligned}$$

Hence, using Theorem 16 with $\eta \leq \frac{1}{2}$, together with $\ln(1+x) \leq x$:

$$\ln q - g(q) \leq \frac{\eta}{2}(Q-1) + \eta^2.$$

Finally, by linearity of g we have $g(\sum_i a_i q_i) = \sum_i a_i g(q_i)$ and thus since $\ln_\eta(q) \geq g(q)$:

$$\ln \sum_i a_i q_i \leq \sum_i a_i g(q_i) + \frac{\eta}{2}(Q-1) + \eta^2 \leq \sum_i a_i \ln_\eta(q_i) + \frac{\eta}{2}(Q-1) + \eta^2.$$

Substituting $\ln_\eta$ and Q by their definitions finishes the proof. ■

Proof (Lemma 4) The beginning of the proof matches that of Lemma 19, and thus we start from Eq. (21). Since $\ln(1 + \bar{\eta} Q) \geq \frac{\bar{\eta} Q}{1 + \bar{\eta} Q} = \frac{\eta Q}{1 - \eta + \eta Q}$:

$$\begin{aligned} \ln \hat{q} - g(\hat{q}) &\leq \ln(1 - \eta + \eta Q) - \frac{1}{\eta} \ln(1-\eta) - 1 \\ &\leq \ln \left(1 + \frac{\eta}{1-\eta} Q\right) - \underbrace{\left(\frac{1}{\eta} - 1\right)}_{= \frac{1-\eta}{\eta}} \ln(1-\eta) - 1 \end{aligned}$$

and since from Lemma 13 $-\ln(1-\eta) \leq \frac{\eta}{1-\eta} = \bar{\eta}$:

$$\ln \hat{q} - g(\hat{q}) \leq \ln(1 + \bar{\eta} Q).$$

By linearity of g, we have $g(\sum_i a_i q_i) = \sum_i a_i g(q_i)$ and thus:

$$\ln \sum_i a_i q_i \leq \sum_i a_i g(q_i) + \ln(1 + \bar{\eta} Q) \leq \sum_i a_i \ln_\eta(q_i) + \ln(1 + \bar{\eta} Q)$$

where the last inequality follows from $\ln_\eta(q) \geq g(q)$. ■

Appendix C. Proofs for the main results

Proof (Theorem 5) Using $\ln(x) \leq x - 1$ followed by Lemma 17 and the definition of the update rule Eq. (7) leads to:

$$\begin{aligned}
\sum_{t=1}^{T} \ln \frac{A_t}{M_t} &\leq \sum_t \left(\frac{A_t}{M_t} - 1 \right) = \sum_t \sum_i a^i \left(\frac{p_t^i}{M_t} - 1 \right) \\
&\leq \sum_t \sum_i a^i \left(\frac{1}{\eta} \ln \left(1 - \eta + \eta \frac{p_t^i}{M_t} \right) + \frac{\eta}{1-\eta} \left(\frac{p_t^i}{M_t} - 1 \right)^2 \right) \\
&= \sum_t \sum_i a^i \left(\frac{1}{\eta} \ln \frac{w_{t+1}^i}{w_t^i} + \frac{\eta}{1-\eta} \left(\frac{p_t^i}{M_t} - 1 \right)^2 \right) \\
&= \frac{1}{\eta} \sum_i a^i \ln \frac{w_{T+1}^i}{w_1^i} + \frac{\eta}{1-\eta} \sum_i a^i \sum_t \left(\frac{p_t^i}{M_t} - 1 \right)^2 \\
&\leq \frac{1}{\eta} \ln N + \frac{\eta}{1-\eta} \max_i \sum_t \left(\frac{p_t^i}{M_t} - 1 \right)^2 \\
&= \frac{1-\eta}{\eta} \ln N + \frac{\eta}{1-\eta} \max_i \sum_t \left(\frac{p_t^i}{M_t} - 1 \right)^2 + \ln N \,.
\end{aligned}$$

The result is completed by substituting the definitions. ■

Proof (Theorem 6) For the first entry in the minimum we use the fact that $\log x \leq x - 1$:

$$\sum_{t=1}^{T} \ln \frac{A_t}{M_t} \leq \sum_t \left(\frac{A_t}{M} - 1 \right) = \sum_t \sum_i a^i \left(\frac{p_t^i}{M_t} - 1 \right) \leq \sum_t \max_i \left(\frac{p_t^i}{M_t} - 1 \right) .$$

For the other terms we use Lemma 19 to obtain:

$$\begin{aligned}
\ln \frac{A_t}{M_t} &= \ln \left(\sum_i a^i \frac{p_t^i}{M_t} \right) \leq \sum_i a^i \frac{1}{\eta} \ln \left(1 - \eta + \eta \frac{p_t^i}{M_t} \right) + \frac{\eta}{2} \max_i \left(\frac{p_t^i}{M_t} - 1 \right) + \eta^2 \\
&= \sum_i a^i \frac{1}{\eta} \ln \frac{w_{t+1}^i}{w_t^i} + \frac{\eta}{2} \max_i \left(\frac{p_t^i}{M_t} - 1 \right) + \eta^2 \,.
\end{aligned}$$

Therefore

$$\sum_{t=1}^{T} \ln \frac{A_t}{M_t} \leq \frac{1}{\eta} \ln N + \frac{\eta}{2} \sum_{t=1}^{T} \max_i \left(\frac{p_t^i}{M_t} - 1 \right) + \eta^2 T.$$

■

Proof (Theorem 11) As for the proof of Theorem 10 we start from Lemma 9:

$$\ln \frac{A_{1:T}}{M_{1:T}} \leq \frac{1}{\eta_{T+1}} \mathrm{RE}(a \| w_1) + \ln \frac{\eta_1}{\eta_{T+1}} + \sum_{t=1}^{T} \frac{\eta_t}{1-\eta_t} + \sum_{t=1}^{T} \max_{i \leq N} \ln \frac{w_{t+1}^i}{w_t^i}$$

but instead of complementing the missing terms for each expert from $t = 1$ to T (which would lead to $O(m\sqrt{T})$ if all $T_i \approx T$) we complement using the loss injection property of Eq. (18) only for expert i from T_i to T and rely on the restarting property of Eq. (16) to start with a high enough weight $w_{T_i}^i \approx 1/t$. Telescoping the series, we obtain:

$$\ln \frac{A_{1:T}}{M_{1:T}} \leq \frac{1}{\eta_{T+1}} \ln N + \sum_{i \in \mathcal{M}^*} \left[\ln \frac{w_{T+1}^i}{w_{T_i}^i} + \ln \frac{\eta_{T_i}}{\eta_T} + \sum_{t=T_i}^{T} \frac{\eta_t}{1-\eta_t} \right].$$

Using the restarting property of Eq. (16):

$$\begin{aligned}
\sum_{i \in \mathcal{M}^*} \ln \frac{w_{T+1}^i}{w_{T_i}^i} &\leq \sum_{i \in \mathcal{M}^*} \ln \frac{w_{T+1}^i}{w_1^i \left(1 - \frac{\eta_{T_i}}{\eta_{T_i - 1}}\right)} = \sum_{i \in \mathcal{M}^*} \ln \frac{w_{T+1}^i}{w_1^i} - \ln \left(1 - \frac{\eta_{T_i}}{\eta_{T_i-1}}\right) \\
&\leq m \ln \frac{N}{m} - \sum_{i \in \mathcal{M}^*} \ln \left(1 - \frac{\eta_{T_i}}{\eta_{T_i-1}}\right),
\end{aligned}$$

$$\begin{aligned}
-\sum_{i \in \mathcal{M}^*} \ln \left(1 - \frac{\eta_{T_i}}{\eta_{T_i-1}}\right) &= -\sum_{i \in \mathcal{M}^*} \ln \left(1 - \sqrt{\frac{m_{T_i-1}(T_i-1)}{m_{T_i} T_i}}\right) \\
&\leq -\sum_{i \in \mathcal{M}^*} \ln \left(1 - \sqrt{\frac{T_i - 1}{T_i}}\right) = -\sum_{i \in \mathcal{M}^*} \ln \left(1 - \sqrt{1 - \frac{1}{T_i}}\right) \\
&\leq \sum_{i \in \mathcal{M}^*} \ln(2T_i) = m \ln 2 + \sum_{i \in \mathcal{M}^*} \ln T_i
\end{aligned}$$

where we used $\sqrt{1-x} \leq 1 - x/2$. For the next term, remember that $\mathcal{M}_{T_i}^*$ does not yet include the expert i, which will be added at $T_i + 1$:

$$\begin{aligned}
\sum_{i \in \mathcal{M}^*} \ln \frac{\eta_{T_i}}{\eta_T} &\leq \sum_{i \in \mathcal{M}^*} \ln \sqrt{\frac{mT}{m_{T_i} T_i}} = \tfrac{1}{2}(m \ln m - \ln((m-1)!) + \tfrac{1}{2} \sum_{i \in \mathcal{M}^*} \ln \frac{T}{T_i} \\
&\leq \tfrac{1}{2}(m + \ln m) + \frac{m}{2} \ln T - \tfrac{1}{2} \sum_{i \in \mathcal{M}^*} \ln T_i
\end{aligned}$$

where we used $\ln((m-1)!) = \ln(m!) - \ln m$ and $\ln(m!) \geq m\ln(m) - m$. For the next term, we have:

$$\begin{aligned}
\sum_{i\in\mathcal{M}^*}\sum_{t=T_i}^{T}\frac{\eta_t}{1-\eta_t} &= \sum_{i\in\mathcal{M}^*}\sum_{t=T_i}^{T}\eta_t + \underbrace{\frac{\eta_t^2}{1-\eta_t}}_{\leq 2\eta_t^2} = \sum_{t=1}^{T} m_{t+1}(\eta_t + 2\eta_t^2) \\
&= \sum_{t=1}^{T} m_t(\eta_t + 2\eta_t^2) + \sum_{i\in\mathcal{M}^*}\eta_{T_i} + 2\eta_{T_i}^2 .
\end{aligned}$$

For the rightmost term, we take the worst case of largest learning rates, that is for $m_t = t$ up to $t = m$:

$$\begin{aligned}
\sum_{i\in\mathcal{M}^*}\eta_{T_i} + 2\eta_{T_i}^2 &\leq \sum_{t=1}^{m}\sqrt{\frac{\ln N}{2t^2}} + \frac{\ln N}{t^2} \leq \sqrt{\tfrac{1}{2}\ln N}(1+\ln m) + 2\ln N\,, \\
\sum_{t=1}^{T} m_t(\eta_t + 2\eta_t^2) &= \sum_{t=1}^{T}\sqrt{\frac{m_t\ln N}{2t}} + \ln N\frac{1}{t} \leq \sum_{t=1}^{T}\sqrt{\frac{m\ln N}{2t}} + \ln N\frac{1}{t} \\
&\leq \sqrt{2m(T+1)\ln N} + (1+\ln T)\ln N.
\end{aligned}$$

Putting it all together we have:

$$\begin{aligned}
\ln\frac{A_{1:T}}{M_{1:T}} &\leq 2\sqrt{2m(T+1)\ln N} + \left(\frac{m}{2} + \ln N\right)\ln T + \tfrac{1}{2}\sum_{i\in\mathcal{M}^*}\ln T_i + m\ln\frac{N}{m} \\
&\quad + m\ln 2 + \tfrac{1}{2}m + \tfrac{1}{2}\ln m + \sqrt{\tfrac{1}{2}\ln N}(1+\ln m) + 3\ln N \\
&\leq 2\sqrt{2m(T+1)\ln N} + (m+\ln N)\ln T + m\ln\frac{N}{m} \\
&\quad + 1.2m + \sqrt{\tfrac{1}{2}\ln N}(1+\ln m) + 3.5\ln N
\end{aligned}$$

which concludes the proof. ■

Proof (Theorem 12) The beginning of the proof is similar to that of Theorem 10, using first Lemma 4:

$$\begin{aligned}
\ln\frac{A_{1:T}}{M_{1:T}} &= \sum_{k=1}^{K}\ln\frac{A^k_{T_k:T_{k'}}}{M_{T_k:T_{k'}}} \\
&\leq \sum_{k=1}^{K}\sum_{t=T_k}^{T_{k'}}\left[\sum_{i=1}^{N} a_k^i\frac{1}{\eta_t}\ln\left(1-\eta_t+\eta_t\frac{p_t^i}{M_t}\right)\right] + \max_i\ln\left(1+\bar{\eta}_t\frac{p_t^i}{M_t}\right) \\
&= \underbrace{\sum_{t=1}^{T}\max_i\ln\left(1+\bar{\eta}_t\frac{p_t^i}{M_t}\right)}_{(A)} + \sum_{k=1}^{K}\underbrace{\sum_{t=T_k}^{T_{k'}}\sum_{i=1}^{N} a_k^i\frac{1}{\eta_t}\ln\left(1-\eta_t+\eta_t\frac{p_t^i}{M_t}\right)}_{(B)} .
\end{aligned}$$

For (A), we first apply the same transformation as in Eq. (20), then we repeatedly use the loss injection property of Eq. (18) and the telescoping property of Eq. (17) as in the proof of Theorem 10. It can be shown that $\max_{N\geq 2, t\geq 1}\eta_t \leq 3/5$, hence:

$$\begin{aligned}
(A) &\leq N\ln\frac{\eta_1}{\eta_{T+1}} + N\sum_{t=1}^{T}\eta_t + \frac{\eta_t^2}{1-\eta_t} \\
&\leq N\ln\frac{\ln(4)\sqrt{T+1}}{\ln(T+4)} + \sqrt{\tfrac{1}{2}N\ln N}\ln(T+3)\sum_{t=1}^{T}\frac{1}{\sqrt{t}} + \frac{5}{2}\frac{\ln N}{2N}(\ln(T+3))^2\sum_{t=1}^{T}\frac{1}{t} \\
&\leq \frac{N}{2}\ln(T+1) + \sqrt{2N\ln N}\ln(T+3)\sqrt{T} + \frac{5}{4}\frac{\ln N}{N}(\underbrace{\ln(T+3)}_{\leq 1+\ln T, \forall T\geq 2})^2(1+\ln T) \\
&\leq \frac{N}{2}\ln(T+1) + \sqrt{2TN\ln N}\ln(T+3) + \frac{5}{4}\frac{\ln N}{N}(1+\ln T)^3
\end{aligned}$$

For (B), using the $1/\eta$-telescoping property of Eq. (19) and then telescoping the series gives:

$$(B) \leq \sum_{i=1}^{N} a_k^i\left(\frac{1}{\eta_{T_{k+1}}}\ln\frac{w^i_{T_{k+1}}}{w^i_1} - \frac{1}{\eta_{T_k}}\ln\frac{w^i_{T_k}}{w^i_1}\right).$$

Now, with the restarting property of Eq. (16), $\ln\frac{w^i_{T_k}}{w^i_1} \geq \ln\left(1-\frac{\eta_{T_k}}{\eta_{T_k-1}}\right)$ together with Lemma 18, and also with $\sum_{i=1}^{N} a_k^i \ln\frac{w^i_{T_{k+1}}}{w^i_1} \leq \ln N$ we have:

$$\begin{aligned}
(B) &\leq \frac{1}{\eta_{T_{k+1}}}\ln N - \frac{1}{\eta_{T_k}}\ln\left(1-\frac{\eta_{T_k}}{\eta_{T_k-1}}\right) \leq \frac{1}{\eta_{T+1}}\ln N + \frac{1}{\eta_{T+1}}(\ln(T)+1.6) \\
&\leq \sqrt{\frac{2N(T+1)}{\ln N}}\left(1+\frac{\ln N}{\ln(T+4)} + \underbrace{1.6/\ln(T+4)}_{\leq 1}\right) \leq \sqrt{2(T+1)N\ln N}\left(\frac{2}{\ln N}+\frac{1}{\ln T}\right),
\end{aligned}$$

$$\begin{aligned}
\ln\frac{A_{1:T}}{M_{1:T}} &\leq \sqrt{2(T+1)N\ln N}\left(\ln(T+3) + K\left(\frac{2}{\ln N}+\frac{1}{\ln T}\right)\right) \\
&\quad + \frac{5}{4}\frac{\ln N}{N}(1+\ln T)^3 + \frac{N}{2}\ln(T+1)
\end{aligned}$$

which was to be proven. ■

Appendix D. Failure of EG and OGD

Proof (Theorem 2) We start by proving the claim for EG. The theorem will follow by considering two examples. For the first, let $p_t^a = 0$ and $p_t^b = 1$ for all t. Then according to the EG update rule in Eq. (8):

$$w_t^a \propto w_1^a \qquad \text{and} \qquad w_t^b \propto w_1^b \exp\left(\eta\sum_{s=1}^{t}\frac{1}{w_s^b}\right). \tag{22}$$

It is easy to see that $w_s^b \in [1/2, 1]$, which means that

$$w_t^b = \frac{\exp\left(\eta \sum_{s=1}^t \frac{1}{w_s^b}\right)}{1 + \exp\left(\eta \sum_{s=1}^t \frac{1}{w_s^b}\right)} \le \frac{\exp(2\eta t)}{1 + \exp(2\eta t)} .$$

The best mixture in hindsight in this case assigns all mass to the second expert and suffers no loss. Therefore the regret

$$\mathcal{R}_T = \sum_{t=1}^T \ln\left(\frac{1}{w_t^b}\right) \ge \sum_{t=1}^{\min\{1/\eta, T\}} \ln\left(\frac{1+\exp(2)}{\exp(2)}\right) \ge \frac{1}{10} \min\left\{\frac{1}{\eta}, T\right\} ,$$

which proves the result for small learning rates $\eta \le T^{\varepsilon - 1}$. From now on suppose that T is reasonably large and $\eta \ge T^{\varepsilon-1}$. Now we consider a different sequence of expert predictions. Let the first expert be a poor predictor for the first $T/2$ rounds, and subsequently let the prediction quality of each expert alternate. Formally,

$$p_t^a = \begin{cases} 0 & \text{if } t \le T/2 \\ 1 & \text{if } t > T/2 \text{ and } t \text{ is even} \\ 0 & \text{otherwise} . \end{cases} \qquad p_t^b = \begin{cases} 1 & \text{if } t \le T/2 \\ 0 & \text{if } t > T/2 \text{ and } t \text{ is even} \\ 1 & \text{otherwise} . \end{cases}$$

Notice that this is the same sequence as considered in the first part until $t = T/2$. Therefore by Eq. (22) for $t = T/2 + 1$ we have

$$w_{T/2+1}^a = \left(1 + \exp\left(\eta \sum_{s=1}^{T/2} \frac{1}{w_s^b}\right)\right)^{-1} \le \exp(-\eta T/2) .$$

This means that in a single round the predictor suffers loss $-\log \exp(-\eta T/2) = \eta T/2$, which may already be quite large. But there are still many rounds to go and things do not get better. In round $T/2+2$ we have

$$\begin{aligned} w_{T/2+2}^b &= \frac{w_{T/2+1}^b}{w_{T/2+1}^b + w_{T/2+1}^a \exp\left(\eta / w_{T/2+1}^a\right)} \le \frac{\exp\left(-\eta / w_{T/2+1}^a\right)}{w_{T/2+1}^a} \\ &\le \frac{\exp\left(-\eta \exp(\eta T/2)\right)}{\exp(-\eta T/2)} \le \exp(-\eta T/2) \end{aligned} \tag{23}$$

and so on. Therefore the loss of the EG algorithm over the final $T/2$ rounds is at least $\eta T^b/4 = \Omega(T^{1+\varepsilon})$. For this sequence the loss of the best mixture in hindsight is $O(T)$ and hence the regret of EG is at least $\Omega(T^{1+\varepsilon})$, which completes the proof for EG. Moving to gradient descent. We use a similar example where in rounds $t < T$ the first expert has $p_t^a = 0$ and the second has $p_t^b = 1$. An easy calculation shows that

$$w_{t+1}^b = \max\left\{1,\, w_t^b + \frac{\eta}{2w_t^b}\right\} \ge \max\left\{1,\, w_t^b + \frac{\eta}{2}\right\} .$$

Therefore since $w_1^b = 1/2$, if $T > 1 + 1/\eta$, then $w_T^b = 1$. In this case let $p_T^a = 1$ and $p_T^b = 0$, which leads to infinite regret. On the other hand if $T \le 1+1/\eta$, then let $p_T^a = 0$ and $p_T^b = 1$ so that the minimum loss in hindsight vanishes. Therefore the regret of OGD is at least

$$\mathcal{R}_T = \sum_{t=1}^{T} \ln\left(\frac{1}{w_t^b}\right) \ge \sum_{t=1}^{T/2} \ln\left(\frac{1}{1/2 + \eta t}\right) = \Omega(T)\,,$$

where we used the fact that $w_t^b \ge 1/2$ so that $\max\{1,\, w_t^b + \eta/(2w_t^b)\} \le \max\{1,\, w_t^b + \eta\}$. ■

Remark 20 *Notice that the second inequality in Eq. (23) is rather loose when η is large. In fact for learning rates $\eta = O(T^{\varepsilon-1})$ one can show the regret of EG grows super-exponentially.*

Appendix E. A multi-learning-rate Soft-Bayes

ML-Prod (Gaillard et al., 2014) was designed for [0,1] losses for linear optimization. It uses one learning rate per expert to be able to track the best of them with a loss that does not depend on the other experts. It is not clear how to use the gradient trick to transform ML-Prod to learn a convex combination of the experts in the logarithmic loss, but we can still provide a very similar result. Using fixed learning rates η^i, we define the mixture as:

$$\begin{aligned}
M_{1:T} &:= \sum_{i=1}^{N} w_1^i \prod_{t=1}^{T}((1-\eta^i)M_t + \eta^i p_t^i) \\
M_t &= \frac{M_{1:t}}{M_{<t}} = \frac{\sum_i w_t^i \eta^i p_t^i}{\sum_i w_t^i \eta^i} \\
w_{t+1}^i &:= w_1^i \frac{\prod_{t=1}^{T}((1-\eta^i)M_t + \eta^i p_t^i)}{M_{1:T}} = w_t^i \left(1 - \eta^i + \eta^i \frac{p_t^i}{M_t}\right)
\end{aligned}$$

where the equation for M_t follows from $M_t = \sum_i w_t^i[(1-\eta^i)M_t + \eta^i p_t^i]$ and algebra. For time-dependent learning rates, we need to apply the online correction rule Eq. (14) to the update of the weights, which gives the following:

$$M_t := \frac{M_{1:t}}{M_{<t}} = \frac{\sum_i w_t^i \eta_t^i p_t^i}{\sum_i w_t^i \eta_t^i} \tag{24}$$

$$w_{t+1}^i := w_t^i \left(1 - \eta_t^i + \eta_t^i \frac{p_t^i}{M_t}\right) \frac{\eta_{t+1}^i}{\eta_t^i} + \left(1 - \frac{\eta_{t+1}^i}{\eta_t^i}\right) w_1^i\,. \tag{25}$$

Theorem 21 *For the algorithm defined by Eqs. (24) and (25), with a sequence $\eta_t^i \in (0,1)$ of monotonically descreasing learning rates, we have the following regret guarantee against the best fixed convex combination of the experts A in hindsight such that $A_t := \sum_{i=1}^N a^i p_t^i$:*

$$\ln\frac{A_{1:T}}{M_{1:T}} \le \sum_{i=1}^{N} a^i \left[\frac{1}{\bar\eta_{T+1}^i} \ln\frac{1}{w_1^i} + \sum_{t=1}^{T} \bar\eta_t^i \left(\frac{p_t^i}{M_t} - 1\right)^2 + \ln\frac{1}{w_1^i}\right], \qquad \textit{with} \quad \bar\eta_t^i := \frac{\eta_t^i}{1-\eta_t^i}.$$

Proof Using $\ln(x) \leq x-1$, then Lemma 17 on the second line, the $1/\eta$-telescoping property of Eq. (19) on the third line, and telescoping on the fourth line we have:

$$\begin{aligned}
\sum_{t=1}^{T} \ln \frac{A_t}{M_t} &\leq \sum_t \left(\frac{A_t}{M_t} - 1\right) = \sum_t \sum_i a^i \left(\frac{p_t^i}{M_t} - 1\right) = \sum_i a^i \sum_t \left(\frac{p_t^i}{M_t} - 1\right) \\
&\leq \sum_i a^i \sum_t \frac{1}{\eta_t^i} \ln\left(1 - \eta_t^i + \eta_t^i \frac{p_t^i}{M_t}\right) + \frac{\eta_t^i}{1-\eta_t^i}\left(\frac{p_t^i}{M_t} - 1\right)^2 \\
&\leq \sum_i a^i \sum_t \left(\frac{1}{\eta_{t+1}^i} \ln \frac{w_{t+1}^i}{w_1^i} - \frac{1}{\eta_t^i} \ln \frac{w_{t+1}^i}{w_1^i}\right) + \frac{\eta_t^i}{1-\eta_t^i}\left(\frac{p_t^i}{M_t} - 1\right)^2 \\
&= \sum_i a^i \left[\frac{1}{\eta_{T+1}^i} \ln \frac{w_{T+1}^i}{w_1^i} + \sum_{t=1}^{T} \frac{\eta_t^i}{1-\eta_t^i}\left(\frac{p_t^i}{M_t} - 1\right)^2\right] \\
&= \sum_i a^i \left[\frac{1-\eta_{T+1}^i}{\eta_{T+1}^i} \ln \frac{w_{T+1}^i}{w_1^i} + \frac{\eta_{T+1}^i}{1-\eta_{T+1}^i} \sum_{t=1}^{T} \left(\frac{p_t^i}{M_t} - 1\right)^2 + \ln \frac{w_{t+1}^i}{w_1^i}\right] \\
&\leq \sum_i a^i \left[\frac{1-\eta_{T+1}^i}{\eta_{T+1}^i} \ln \frac{1}{w_1^i} + \frac{\eta_{T+1}^i}{1-\eta_{T+1}^i} \sum_{t=1}^{T} \left(\frac{p_t^i}{M_t} - 1\right)^2 + \ln \frac{1}{w_1^i}\right] \\
&\leq \max_i \left[\frac{1-\eta_{T+1}^i}{\eta_{T+1}^i} \ln \frac{1}{w_1^i} + \frac{\eta_{T+1}^i}{1-\eta_{T+1}^i} \sum_{t=1}^{T} \left(\frac{p_t^i}{M_t} - 1\right)^2 + \ln \frac{1}{w_1^i}\right].
\end{aligned}$$

■

Appendix F. Sequence prediction with experts with disjoint supports

In this section we consider that the experts have disjoint supports, that is, for any observation exactly one expert predicts it with positive probability:[4] $\forall t \in [1..T], |\{i \in [1..N] | p_t^i > 0\}| = 1$. This happens in particular if the experts are designed so that each expert i predicts the symbol of index i, that is $\forall t, i : p_t^i(x_t = i) = 1$ when considering that $\mathcal{X} = [1..N]$.

In this setting, we show that the Soft-Bayes rule with a learning rate η_t of $\frac{1}{t+c}$ recovers exactly some well-known density estimators such as Laplace's rule of succession and the minmax-optimal KT estimator.

Let i_t be the index of the model that places positive probability for the current observation x_t at time t. Then $M_t = \sum_i w_t^i p_t^i = w_t^{\hat{i}_t} p_t^{\hat{i}_t}$.

Let $n_t^i := \sum_{s=1}^{t} [\![\hat{i}_s = i]\!]$ be the number of times up to t where the expert i is correct.

Then we have the following property.

Theorem 22 *If the experts have disjoint supports and uniform prior $w_1^i = 1/N$, then using a learning rate $\eta_t := \frac{1}{t+c}$ makes mixture predict*

$$\forall t : M_{t+1}(x_{t+1} = i) = \frac{n_t^i + \frac{c}{N}}{t+c} p_{t+1}^i.$$

[4] But a single expert can still place positive probability over several observations, as long as there is no overlap with any other expert.

Proof We proceed by induction on the weights:

$$\begin{aligned}
w_{t+1}^{\hat{i}_t} &= w_t^{\hat{i}_t}(1-\eta_t+\eta_t\frac{p_t^{\hat{i}_t}}{M_t}) = w_t^i(1-\eta_t)+\eta_t \\
\forall i \neq \hat{i}_t : w_{t+1}^i &= w_t^i(1-\eta_t) \\
\text{that is } \forall i : w_{t+1}^i &= w_t^i(1-\eta_t)+\eta_t[\![\hat{i}_t = i]\!]
\end{aligned}$$

Now with $\eta_t = \frac{1}{t+c}$, observe[5] that $\frac{\eta_t}{1-\eta_t} = \frac{1}{t-1+c} = \eta_{t-1}$. Then

$$\begin{aligned}
w_{t+1}^i &= w_1^i(1-\eta_t)+\eta_t[\![\hat{i}_t = i]\!] \\
\frac{1}{\eta_t}w_{t+1}^i &= \frac{1-\eta_t}{\eta_t}w_1^i + [\![\hat{i}_t = i]\!] \\
&= \frac{1}{\eta_{t-1}}w_t^i + [\![\hat{i}_t = i]\!] \\
&= \frac{1}{\eta_0}w_1^i + \sum_{s=1}^{t}[\![\hat{i}_s = i]\!] \quad \text{(by induction)} \\
\frac{1}{t+c}w_{t+1}^i &= \frac{c}{N} + n_t^i, \\
\text{hence} \quad w_{t+1}^i &= \frac{n_t^i + \frac{c}{N}}{t+c}
\end{aligned}$$

which with $M_{t+1} = w_{t+1}^i p_{t+1}^i$ for $i = \hat{i}_{t+1}$ proves the claim. ∎

In particular, for experts such that $p_t^i \in \{0,1\}$ and compared to the best constant convex combination of the experts in hindsight (still with disjoint support),

- setting $c = 1$ recovers Perks' estimator (Perks, 1947; Hutter, 2013), with a regret of $O(m \ln T)$ where m is the number of experts that make at least one good prediction,
- setting $c = N$ recovers Laplace's rule of succession, with a regret of $O(N \ln \frac{T}{N})$,
- setting $c = N/2$ recovers the KT estimator (Krichevsky and Trofimov, 1981), with a regret of $O(\frac{N}{2}\ln T)$.

See (Hutter, 2013) for more details and comparison of these estimators.

We can draw an interesting parallel between these estimators and the different learning rates for competing against a fixed combination of the experts (with non-disjoint supports).

Indeed Perks' estimator with learning rate $\eta_t = \frac{1}{t+1}$ is a sparse estimator: It pays a cost of $\log t$ each time a symbol is seen for the first time, just like a learning rate of $\eta_t = \frac{1}{\sqrt{t}}$ pays a cost of $\sqrt{t}$ for convex combinations when an expert is good for the first time. The KT estimator with a learning rate of $\frac{1}{t+N/2}$ minimizes the worst case where all symbols must be introduced, similarly to a learning rate of $\frac{1}{\sqrt{tN}}$ for convex combinations.

References

Peter Auer and Claudio Gentile. Adaptive and self-confident on-line learning algorithms. In *Proceedings of COLT'00*, pages 107–117, 2000.

[5] Interestingly, this property exists only for this type of learning rate, and not for example for $\eta_t \propto \frac{1}{\sqrt{t}}$.

Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *J. ACM*, 44(3):427–485, May 1997.

Nicolò Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2-3):321–352, 2007.

Alexey Chernov and Vladimir Vovk. Prediction with expert evaluators' advice. In *Algorithmic Learning Theory*, volume 5809 of *Lecture Notes in Artificial Intelligence*, pages 8–22. Springer, 2009.

Thomas M Cover. Universal portfolios. *Mathematical finance*, 1(1):1–29, 1991.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on machine learning*, pages 272–279. ACM, 2008.

Yoav Freund, Robert E Schapire, Yoram Singer, and Manfred K Warmuth. Using and combining predictors that specialize. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 334–343. ACM, 1997.

Pierre Gaillard, Gilles Stoltz, and Tim van Erven. A second-order bound with excess losses. *Journal of Machine Learning Research, W&CP: COLT*, 35:176–196, 2014.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2):169–192, 2007.

David P Helmbold, Robert E Schapire, Yoram Singer, and Manfred K Warmuth. On-line portfolio selection using multiplicative updates. *Mathematical Finance*, 8(4):325–347, 1998.

Mark Herbster and Manfred K. Warmuth. Tracking the best expert. *Machine Learning*, 32 (2):151–178, August 1998.

Marcus Hutter. Sparse adaptive Dirichlet-multinomial-like processes. *Journal of Machine Learning Research, W&CP: COLT*, 30:432–459, 2013.

Adam Kalai and Santosh Vempala. Efficient algorithms for universal portfolios. *Journal of Machine Learning Research*, 3(Nov):423–440, 2002.

Wouter M. Koolen, Dmitry Adamskiy, and Manfred K. Warmuth. Putting bayes to sleep. In *Advances in Neural Information Processing Systems*, pages 135–143, 2012.

R Krichevsky and V Trofimov. The performance of universal encoding. *IEEE Transactions on Information Theory*, 27(2):199–207, 1981.

E. R. Love. 64.4 Some logarithm inequalities. *The Mathematical Gazette*, 64(427):55–57, 1980.

Haipeng Luo, Alekh Agarwal, Nicolò Cesa-Bianchi, and John Langford. Efficient second order online learning by sketching. In *Advances in Neural Information Processing Systems*, pages 902–910, 2016.

Christopher Mattern. *On Statistical Data Compression*. PhD thesis, Technische Universität Ilmenau, Fakultät für Informatik und Automatisierung, Feb 2016.

Wilfred Perks. Some observations on inverse probability including a new indifference rule. *Journal of the Institute of Actuaries*, 73(2):285–334, 1947.

Amir Sani, Gergely Neu, and Alessandro Lazaric. Exploiting easy data in online optimization. In *Advances in Neural Information Processing Systems*, pages 810–818, 2014.

Joel Veness, Kee Siong Ng, Marcus Hutter, and Michael Bowling. Context tree switching. In *Data Compression Conference (DCC), 2012*, pages 327–336. IEEE, 2012a.

Joel Veness, Peter Sunehag, and Marcus Hutter. On ensemble techniques for AIXI approximation. In *International Conference on Artificial General Intelligence*, pages 341–351. Springer, 2012b.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 928–935, 2003.