

# Efficient Tracking of Large Classes of Experts

András György, *Member, IEEE*, Tamás Linder, *Senior Member, IEEE*, and Gábor Lugosi

November 11, 2018

**Abstract**—In the framework of prediction of individual sequences, sequential prediction methods are to be constructed that perform nearly as well as the best expert from a given class. We consider prediction strategies that compete with the class of switching strategies that can segment a given sequence into several blocks, and follow the advice of a different “base” expert in each block. As usual, the performance of the algorithm is measured by the regret defined as the excess loss relative to the best switching strategy selected in hindsight for the particular sequence to be predicted. In this paper we construct prediction strategies of low computational cost for the case where the set of base experts is large. In particular we provide a method that can transform any prediction algorithm  $\mathcal{A}$  that is designed for the base class into a tracking algorithm. The resulting tracking algorithm can take advantage of the prediction performance and potential computational efficiency of  $\mathcal{A}$  in the sense that it can be implemented with time and space complexity only  $O(n^\gamma \ln n)$  times larger than that of  $\mathcal{A}$ , where  $n$  is the time horizon and  $\gamma \geq 0$  is a parameter of the algorithm. With  $\mathcal{A}$  properly chosen, our algorithm achieves a regret bound of optimal order for  $\gamma > 0$ , and only  $O(\ln n)$  times larger than the optimal order for  $\gamma = 0$  for all typical regret bound types we examined. For example, for predicting binary sequences with switching parameters under the logarithmic loss, our method achieves the optimal  $O(\ln n)$  regret rate with time complexity  $O(n^{1+\gamma} \ln n)$  for any  $\gamma \in (0, 1)$ .

## I. INTRODUCTION

In the on-line (sequential) decision problems considered in this paper, a decision maker (or forecaster) chooses, at each time instant  $t = 1, 2, \dots$ , an action from a set. After each action taken, the decision maker suffers some loss based on the state of the environment and the chosen decision. The general goal of the forecaster is to minimize its cumulative loss. Specifically, the forecaster’s aim is to achieve a cumulative loss that is not much larger than that of the best expert (forecaster) in a reference class  $\mathcal{E}$ , from which the best expert is chosen in hindsight. This problem is known as “prediction

with expert advice.” The maximum excess loss  $R_n$  of the forecaster relative to the best expert is called the (worst-case) cumulative regret, where the maximum is taken over all possible behaviors of the environment and  $n$  denotes the time horizon of the problem. Several methods are known that can compete successfully with different expert classes in the sense that the regret only grows sub-linearly, that is,  $\lim_{n \rightarrow \infty} R_n/n = 0$ . We refer to [1] for a survey.

While the goal in the standard online prediction problem is to perform nearly as well as the best expert in the class  $\mathcal{E}$ , a more ambitious goal is to compete with the best *sequence* of expert predictions that may switch its experts a certain, limited, number of times. This, seemingly more complex, problem may be regarded as a special case of the standard setup by introducing the so-called *meta experts*. A meta expert is described by a sequence of base experts  $(i_1, \dots, i_n) \in \mathcal{E}^n$ , such that at time instants  $t = 1, \dots, n$  the meta expert follows the predictions of the “base” expert  $i_t \in \mathcal{E}$  by predicting  $f_{i_t, t}$ . The complexity of such a meta expert may be measured by  $C = |\{t \in \{1, 2, \dots, n-1\} : i_t \neq i_{t+1}\}|$ , the number of times it changes the base predictor (each such change is called a switch). Note that  $C$  switches partition  $\{1, \dots, n\}$  into  $C + 1$  contiguous segments, on each of which the meta expert follows the predictions of the same base expert. If a maximum of  $m$  changes are allowed and the set of base experts has  $N$  elements, then the class of meta experts is of size  $\sum_{j=0}^m \binom{n-1}{j} N(N-1)^j$ . Since the computational complexity of basic prediction algorithms, such as the exponentially weighted average forecaster, scales with the number of experts, a naive implementation of these algorithms is not feasible in this case. However, several more efficient algorithms have been proposed.

One approach, widely used in the information theory/source coding literature, is based on transition diagrams [2], [3]: A transition diagram is used to define a prior distribution on the switches of the experts, and the starting point of the current segment is estimated using this prior. A transition diagram defines a Markovian model on the switching times: a state of the model describes the “status” of a switch process (corresponding to, e.g., the time when the last switch occurred and the actual time), and the transition diagram defines the transition probabilities among these states. In its straightforward version, at each time instant  $t$ , the performance of an expert algorithm is emulated for all possible segment starting points  $1, \dots, t$ , and a weighted average of the resulting estimates is used to form the next prediction. In effect, this method converts an efficient algorithm to compete with the best expert in a class  $\mathcal{E}$  into one that competes with the best sequence of experts

This research was supported in part by the National Development Agency of Hungary from the Research and Technological Innovation Fund (KTIA-OTKA CNK 77782), the Alberta Innovates Technology Futures, the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Spanish Ministry of Science and Technology grant MTM2009-09063, and the PASCAL2 Network of Excellence under EC grant no. 216886.

The material in this paper was presented in part at the 2012 IEEE International Symposium on Information Theory, Cambridge, MA, USA, July 2012.

A. György is with the Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada T6G 2E8; during part of this work he was with the Machine Learning Research Group, Computer and Automation Research Institute of the Hungarian Academy of Sciences, Budapest, Hungary, (email: gya@szit.bme.hu). T. Linder is with the Department of Mathematics and Statistics, Queen’s University, Kingston, Ontario, Canada K7L 3N6 (email: linder@mast.queensu.ca). G. Lugosi is with ICREA and the Department of Economics, Pompeu Fabra University, Ramon Trias Fargas 25-27, 08005 Barcelona, Spain (email: gabor.lugosi@gmail.com).

with a limited number of changes. The time complexity of the method depends on how complex the prior distribution is, which determines the amount of computation necessary to update the weights in the estimate. Note that a general prior distribution would require exponential computational complexity in the sequence length, while at each time instant the transition diagram model requires computations proportional to the number of achievable states at that time instant. Using a state space that describes the actual time, the time of the last switch, and the number of switches so far, [2] provided a prediction scheme achieving the optimal regret up to an additive constant (for the logarithmic loss), and, omitting the number of switches from the states, a prediction algorithm with optimal regret rate was provided. [3] showed (also for the logarithmic loss) that the transition probabilities in the latter model can be selected so that the resulting prediction scheme achieves the optimal regret rate with the best possible leading constant, and the distributions they use allow computing the weights at time instant  $t$  with  $O(t)$  complexity. As a result, in  $n$  time steps, the time complexity of the best transition-diagram based algorithm is a factor  $O(n)$  times larger than that of the original algorithm that competes with  $\mathcal{E}$ , yielding a total complexity that is quadratic in  $n$ .

For the same problem, a method of linear complexity was developed in [4]. It was shown in [5] that this method is equivalent to an easy-to-implement weighting of the paths in the full transition diagram. Although, unlike transition diagram based methods, the original version of the algorithm of [4] requires an a priori known upper bound on the number of switches, the algorithm can be modified to compete with meta experts with an arbitrary number of switches: a linear complexity variant achieves this goal (by letting its switching parameter  $\alpha$  decrease to zero) at the price of somewhat increasing the regret [6]. A slightly better regret bound can be achieved for the case when switching occurs more often at the price of increasing the computational complexity from linear to  $O(n^{3/2})$  [7], [8] (by discretizing its switching parameter  $\alpha$  to  $\sqrt{n}$  levels).

In another approach, reduced transition diagrams have been used for the logarithmic loss (i.e., lossless data compression) by [9] and by [3] (the latter work considers a probabilistic setup as opposed to the individual sequence setting). Reduced transition diagrams are obtained by restricting some transitions, and consequently, excluding some states from the original transition diagram, resulting in (computationally) simpler models that, however, have less descriptive power to represent switches. An efficient algorithm based on a reduced transition diagram for the general tracking problem was given in [10], while [11] developed independently a similar algorithm to minimize adaptive regret, which is the maximal worst-case cumulative excess loss over any contiguous time segment relative to a constant expert. It is easy to see that algorithms with good adaptive regret also yield good tracking regret.

An important question is how one can compete with meta experts when the base expert class  $\mathcal{E}$  is very large. In such cases special algorithms are needed to compete with experts from the base class even without switching. Such large base classes arise in on-line linear optimization [12], lossless

data compression [13]–[15], the shortest path problem [16], [17], or limited-delay lossy data compression [18]–[20]. Such special algorithms can easily be incorporated in transition-diagram-based tracking methods, but the resulting complexity is quadratic in  $n$  (see, e.g., [3] for such an application to lossless data compression or [21]–[23] for applications to signal processing and universal portfolio selection). If the special algorithms for large base expert classes are combined with the algorithm of [4] to compete with meta experts, the resulting algorithms again have quadratic complexity in  $n$ ; see, e.g., [5], [24] (the main reason for this is that the special implementation tricks used for the large base expert classes, such as dynamic programming, are incompatible with the efficient implementation of the algorithm of [4] for switching experts). The only example we are aware of where efficient tracking algorithms with linear time complexity are available for a meaningful, large class of base experts is the case of online convex programming, where the set of base experts is a finite dimensional convex set and the (time-varying) loss functions are convex [25] (see also the related problem of tracking linear predictors [26]). In this case projected gradient methods (including exponentially weighted average prediction) lead to tracking regret bounds of optimal order. Note that instead of the number of switches, these bounds measure the complexity of the meta experts with the more refined notion of  $L_p$  norms.

In this paper we tackle the complexity issue in competing with meta-experts for large base expert classes by presenting a general method for designing reduced transition diagrams. The resulting algorithm converts any (black-box) prediction algorithm  $\mathcal{A}$  achieving good regret against the base-expert class into one that achieves good tracking and adaptive regret. The advantage of this transition-diagram based approach is that the conversion is independent of the base prediction algorithm  $\mathcal{A}$ , and so some favorable properties of  $\mathcal{A}$  are automatically transferred to our algorithm. In particular, the complexity of our method depends on the base-expert class only through the base prediction algorithm  $\mathcal{A}$ , thus exploiting its potential computational efficiency.<sup>1</sup> Our algorithm unifies and generalizes the algorithms of [9], [11] and our earlier work [10]. This algorithm has an explicit complexity-regret trade-off, covering essentially all such results in the literature. In addition to the (almost) linear complexity algorithms in the aforementioned papers, the parameters of our algorithm can be set to reproduce the methods based on the full transition diagram [2], [3], [21], or the complexity-regret behavior of [7], [8]. Also, our algorithm has regret of optimal order with complexity  $O(n^{1+\gamma} \ln n)$  for any  $\gamma \in (0, 1)$ , while setting  $\gamma = 0$  results in complexity  $O(n \ln n)$  and a regret rate that is only a factor of  $\ln n$  larger than the optimal one (similarly to [9]–[11]).

The rest of the paper is organized as follows. First the online prediction and the tracking problems are introduced in Section II. In Section III-A we describe our general algorithm. Sections III-B and III-C present a unified method for

<sup>1</sup>Other black-box reductions of forecasters for different notions of regret are available in the literature; for example, the conversion of forecasters achieving good external regret to ones achieving good internal regret [27], [28].

### PREDICTION WITH EXPERT ADVICE

For each round  $t = 1, 2, \dots$

- (1) the environment chooses the next outcome  $y_t$  and the expert advice  $\{f_{i,t} \in \mathcal{D} : i \in \mathcal{E}\}$ ; the expert advice is revealed to the forecaster;
- (2) the forecaster chooses the prediction  $\hat{p}_t \in \mathcal{D}$ ;
- (3) the environment reveals the next outcome  $y_t \in \mathcal{Y}$ ;
- (4) the forecaster incurs loss  $\ell(\hat{p}_t, y_t)$  and each expert  $i$  incurs loss  $\ell(f_{i,t}, y_t)$ .

Fig. 1. The repeated game of prediction with expert advice.

the low-complexity implementation of the general algorithm via reduced transition diagrams. Bounds for the performance the algorithm are developed in Section III-D. More explicit bounds are presented for some important special cases in Sections III-E and III-F. The results are extended to the related framework of randomized prediction in Section IV. Some applications to specific examples are given in Section V.

## II. PRELIMINARIES

In this section we review some basic facts about prediction with expert advice, and introduce the tracking problem.

### A. Prediction with expert advice

Let the decision space  $\mathcal{D}$  be a convex subset of a vector space and let  $\mathcal{Y}$  be a set representing the outcome space. Let  $\ell : \mathcal{D} \times \mathcal{Y} \rightarrow \mathbb{R}$  be a loss function, assumed to be convex in its first argument. At each time instant  $t = 1, \dots, n$ , the environment chooses an action  $y_t \in \mathcal{Y}$  and each “expert”  $i$  from a reference class  $\mathcal{E}$  forms its prediction  $f_{i,t} \in \mathcal{D}$ . Then the forecaster chooses an action  $\hat{p}_t \in \mathcal{D}$  (without knowing  $y_t$ ), suffers loss  $\ell(\hat{p}_t, y_t)$ , and the losses  $\ell(f_{i,t}, y_t), i \in \mathcal{E}$  are revealed to the forecaster. (This is known as the full information case and in this paper we only consider this model. In other, well-studied, variants of the problem, the forecaster only receives limited information about the losses.)

The goal of the forecaster is to minimize its cumulative loss  $\hat{L}_n = \sum_{t=1}^n \ell(\hat{p}_t, y_t)$ , which is equivalent to minimizing its excess loss  $\hat{L}_n - \min_{i \in \mathcal{E}} L_{i,n}$  relative to the set of experts  $\mathcal{E}$ , where  $L_{i,n} = \sum_{t=1}^n \ell(f_{i,t}, y_t)$  for all  $i \in \mathcal{E}$ .

Several methods are known that can compete successfully with different expert classes  $\mathcal{E}$  in the sense that the (worst-case) cumulative regret, defined as

$$\begin{aligned} R_n &= \max_{(y_1, \dots, y_n) \in \mathcal{Y}^n} \left( \hat{L}_n - \min_{i \in \mathcal{E}} L_{i,n} \right) \\ &= \max_{(y_1, \dots, y_n) \in \mathcal{Y}^n} \left( \sum_{t=1}^n \ell(\hat{p}_t, y_t) - \min_{i \in \mathcal{E}} \sum_{t=1}^n \ell(f_{i,t}, y_t) \right) \end{aligned}$$

only grows sub-linearly, that is,  $\lim_{n \rightarrow \infty} R_n/n = 0$ . One of the most popular among these is *exponential weighting*. When the expert class  $\mathcal{E}$  is finite or countably infinite, this method assigns, at each time instant  $t$ , the nonnegative weight

$$\pi_{i,t} = \frac{w_i e^{-\eta_t L_{i,t-1}}}{\sum_{j \in \mathcal{E}} w_j e^{-\eta_t L_{j,t-1}}}$$

to each expert  $i \in \mathcal{E}$ . Here  $L_{i,t-1} = \sum_{s=1}^{t-1} \ell(f_{i,s}, y_s)$  is the cumulative loss of expert  $i$  up to time  $t-1$ ,  $\eta_t > 0$  is called the learning parameter, and the  $w_i > 0$  are nonnegative initial weights with  $\sum_{i \in \mathcal{E}} w_i = 1$ , so that  $\sum_{i \in \mathcal{E}} \pi_{i,t} = 1$  (we define  $L_{i,0} = 0$  for all  $i \in \mathcal{E}$ , as well as  $L_0 = 0$ ). The decision chosen by this algorithm is

$$\hat{p}_t = \sum_{i \in \mathcal{E}} \pi_{i,t} f_{i,t} \quad (1)$$

which is well defined since  $\mathcal{D}$  is convex.

In this paper we concentrate on two special types of loss functions: bounded convex and exp-concave. For such loss functions the regret of the exponentially weighted average forecaster is well understood. For example, assume  $\ell$  is convex in its first argument and takes its values in  $[0, 1]$ , and the set of experts is finite with  $|\mathcal{E}| = N$ . If  $\eta_t$  is nonincreasing in  $t$ , then for all  $n$ ,

$$\hat{L}_n \leq \min_i \left\{ L_{i,n} + \frac{1}{\eta_n} \ln \frac{1}{w_i} \right\} + \sum_{t=1}^n \frac{\eta_t}{8}, \quad (2)$$

see [29]. By setting the initial weights to  $w_i = 1/N, i = 1, \dots, N$  and with the choice  $\eta_t = 2\sqrt{\ln N/t}$ , one obtains for all  $n \geq 1$ ,

$$R_n \leq \sqrt{n \ln N}. \quad (3)$$

If, on the other hand, for some  $\eta > 0$  the function  $F(p) = e^{-\eta \ell(p,y)}$  is concave for any fixed  $y \in \mathcal{Y}$  (such loss functions are called *exp-concave*) then, choosing  $\eta_t \equiv \eta$  and  $w_i = 1/N, i = 1, \dots, N$ , one has for all  $n \geq 1$ ,

$$R_n \leq \frac{\ln N}{\eta}. \quad (4)$$

We note that the regret bounds in (2)–(4) do not require a fixed time horizon, that is, they hold simultaneously for all  $n \geq 1$ .

The family of exp-concave loss functions includes, for example, for  $p, y \in [0, 1]$ , the square loss  $\ell(p, y) = (p - y)^2$  with  $\eta \leq 1/2$ , and the relative entropy loss  $\ell(p, y) = y \ln \frac{y}{p} + (1 - y) \ln \frac{1-y}{1-p}$  with  $\eta \leq 1$ . A special case of the latter is the logarithmic loss defined for  $y \in \{0, 1\}$  and  $p \in [0, 1]$  by  $\ell(p, y) = -\mathbb{I}_{y=1} \ln p - \mathbb{I}_{y=0} \ln(1 - p)$ , which plays a central role in data compression. Here and throughout the paper  $\mathbb{I}_B$  denotes the indicator of event  $B$ . We refer to [1] for discussions of these bounds.

### B. The tracking problem

In the standard online prediction problem the goal is to perform as well as the best expert in a given reference class  $\mathcal{E}$ . In this paper we consider the more ambitious goal of competing with a sequence of expert predictions that are allowed to switch between experts. Formally, such a *meta*

*expert* is defined as follows. Fix the time horizon  $n \geq 1$ . A meta expert that changes base experts at most  $C \geq 0$  times can be described by a vector of experts  $a = (i_0, \dots, i_C) \in \mathcal{E}^{C+1}$  and a “transition path”  $T = (t_1, \dots, t_C; n)$  such that  $t_0 := 1 < t_1 < \dots < t_C < t_{C+1} := n + 1$ . For each  $c = 0, \dots, C$ , the meta expert follows the advice of expert  $i_c$  in the time interval  $[t_c, t_{c+1})$ . When the time horizon  $n$  is clear from the context, we will omit it from the description of  $T$  and simply write  $T = (t_1, \dots, t_C)$ . We note that this representation is not unique as the definition does not require that base experts  $i_c$  and  $i_{c+1}$  be different. Any meta expert that can be defined using a given transition path  $T$  is said to follow  $T$ .

The total loss of the meta expert indexed by  $(T, a)$ , accumulated during  $n$  rounds, is

$$L_n(T, a) = \sum_{c=0}^C L_{i_c}(t_c, t_{c+1})$$

where  $L_i(t_1, t_2) = \sum_{t=t_1}^{t_2-1} \ell(f_{i,t}, y_t)$  denotes the loss of expert  $i \in \mathcal{E}$  in the interval  $[t_1, t_2)$ ,  $1 \leq t_1 \leq t_2 \leq n$ . For any  $t \geq 1$ , let  $\mathcal{T}_t$  denote the set of all transition paths up to time  $t$  represented by vectors  $(t_1, \dots, t_C; t)$  with  $1 < t_1 < t_2 < \dots < t_C \leq t$  and  $0 \leq C \leq t$ . For any  $T = (t_1, \dots, t_C) \in \mathcal{T}_n$  and  $t \leq n$  define the truncation of  $T$  at time  $t$  as  $T_t = (t_1, \dots, t_k; t)$ , where  $k$  is such that  $t_k \leq t < t_{k+1}$  (note that  $t \leq n$  guarantees that  $t_{C+1} = n + 1 > t$ , and so  $t_{k+1}$  is well-defined). Furthermore, let  $\tau_t(T) = \tau_t(T_t) = t_k$  denote the last change up to time  $t$ , and let  $C_t(T) = C(T_t) = k$  denote the number of switches up to time  $t$ . A transition path  $T$  with  $C$  switches splits the time interval  $[1, n]$  into  $C + 1$  contiguous segments.

Our goal is to perform nearly as well as the meta-experts, that is, to keep the regret  $\hat{L}_n - L_n(T, a)$  small relative to the meta-experts  $(T, a)$  for all outcome sequences  $y_1, \dots, y_n$ . It is clear that this cannot be done uniformly well for all meta experts; for example, it is obvious that the performance of a meta expert that is allowed to switch experts at each time instant cannot be achieved for all outcome sequences. Indeed, it is known [4], [30] that, for exp-concave loss functions, the worst-case regret of any prediction algorithm relative to the best meta-expert with at most  $C$  switches, selected in hindsight, is at least of the order of  $(C + 1) \log n$ , where the worst-case tracking regret with respect to meta experts with at most  $C$  switches is defined as

$$\max_{y_1, \dots, y_n} \left( \hat{L}_n - \min_{(T, a): C_n(T) = C} L(T, a) \right).$$

Algorithms achieving optimal regret rates are known under general conditions: for general convex loss functions and a finite number of base experts, a tracking regret of order  $(C(T) + 1) \sqrt{n \ln n}$  (or  $\sqrt{(C + 1)n \ln n}$  if  $C$  is known in advance) can be achieved [4], [5], [24], while the  $O((C + 1) \ln n)$  lower bound is achievable in case of exp-concave loss functions and a finite number of experts [2]–[4], [6], [21], or when the base experts form a convex subset of a finite dimensional linear space [31].

We will also consider the related notion of *adaptive regret*

$$R_n^a = \max_{t \leq t'} \max_{y_t, y_{t+1}, \dots, y_{t'}} \left( \sum_{\tau=t}^{t'} \ell(\hat{p}_\tau, y_\tau) - \min_{i \in \mathcal{E}} \sum_{\tau=t}^{t'} \ell(f_{i,\tau}, y_\tau) \right)$$

introduced in [31] and [11], which is the maximal worst-case cumulative excess loss over any contiguous time segment relative to a constant expert. Minimizing the tracking and the adaptive regret are similar problems. In fact, one can show that the FLH1 algorithm of [31] developed to minimize the adaptive regret and a dynamic version of the fixed-share algorithm of [4] introduced by [6] to minimize the tracking regret are identical. Furthermore, any algorithm with small adaptive regret also enjoys small tracking regret, since the regret, in  $n$  time steps, relative to a meta expert that can switch the base expert  $C$  times can be bounded by  $(C + 1)R_n^a$ . Although tracking regret bounds do not immediately yield bounds on the adaptive regret (since the regret on a time segment may be negative), it is usually straightforward to modify the proofs for tracking regret to obtain bounds on the adaptive regret; see, e.g., the proof of Theorem 2.

### III. A REDUCED COMPLEXITY TRACKING ALGORITHM

#### A. A general tracking algorithm

Here we introduce a general tracking method which forms the basis of our reduced complexity tracking algorithm. Consider an on-line forecasting algorithm  $\mathcal{A}$  that chooses an element of the decision space depending on the past outcomes and the expert advices according to the protocol described in Figure 1. Suppose that for all  $n$  and possible outcome sequences of length  $n$ ,  $\mathcal{A}$  satisfies a regret bound

$$R_n \leq \rho_{\mathcal{E}}(n) \quad (5)$$

with respect to the base expert class  $\mathcal{E}$ , where  $\rho_{\mathcal{E}} : [0, \infty) \rightarrow [0, \infty)$  is a nondecreasing and concave function with  $\rho_{\mathcal{E}}(0) = 0$ . These assumptions on  $\rho_{\mathcal{E}}$  are usually satisfied by the known regret bounds for different algorithms, such as the bounds (3) and (4) (with defining  $\rho_{\mathcal{E}}(0) = 0$  in the latter case). Suppose  $1 \leq t_1 < t_2 \leq n$  and an instance of  $\mathcal{A}$  is used for time instants  $t \in [t_1, t_2) := \{t_1, \dots, t_2 - 1\}$ , that is, algorithm  $\mathcal{A}$  is run on data obtained in the segment  $[t_1, t_2)$ . The accumulated loss of  $\mathcal{A}$  during this period will be denoted by  $L_{\mathcal{A}}(t_1, t_2)$ . Then (5) implies

$$L_{\mathcal{A}}(t_1, t_2) - \min_{i \in \mathcal{E}} L_i(t_1, t_2) \leq \rho_{\mathcal{E}}(t_2 - t_1).$$

Running algorithm  $\mathcal{A}$  on a transition path  $T = (t_1, \dots, t_C; n)$  means that at the beginning of each segment of  $T$  (at time instants  $t_c$ ) we restart  $\mathcal{A}$ ; this algorithm will be denoted in the sequel by  $(\mathcal{A}, T)$ . Denote the output of this algorithm at time  $t$  by  $f_{\mathcal{A},t}(T) = f_{\mathcal{A},t}(\tau_t(T))$ . This notation emphasizes the fact that, since  $\mathcal{A}$  is restarted at the beginning of each segment of  $T$ , the output of  $(\mathcal{A}, T)$  at time  $t$  is influenced by  $T$  only through  $\tau_t(T)$ , the beginning of the segment that includes  $t$ . The loss of algorithm  $(\mathcal{A}, T)$  up to time  $n$  is

$$L_n(\mathcal{A}, T) = \sum_{c=0}^C L_{\mathcal{A}}(t_c, t_{c+1}).$$

As most tracking algorithms, our algorithm will use weight functions  $w_t : \mathcal{T}_t \rightarrow [0, 1]$  satisfying

$$\sum_{T_t \in \mathcal{T}_t} w_t(T_t) = 1 \text{ and } w_t(T_t) = \sum_{T'_{t+1} \in \mathcal{T}_{t+1}: T'_t = T_t} w_{t+1}(T'_{t+1}) \quad (6)$$

for all  $t = 1, 2, \dots$  and  $T \in \mathcal{T}$ . Thus each  $w_t$  is a probability distribution on  $\mathcal{T}_t$  such that the family  $\{w_t; t = 1, \dots, n\}$  is consistent. To simplify the notation, we formally define  $T_0$  as the “empty transition path”  $\mathcal{T}_0 := \{T_0\}$ ,  $L_0(\mathcal{A}, T_0) := 0$ , and  $w_0(T_0) := 1$ .

We say that  $\hat{T} \in \mathcal{T}_n$  covers  $T \in \mathcal{T}_n$  if the change points of  $T$  are also change points of  $\hat{T}$ . Note that if  $\hat{T}$  covers  $T$ , then any meta expert that follows transition path  $T$  also follows transition path  $\hat{T}$ . We say that  $w_n$  covers  $\mathcal{T}_n$  if for any  $T \in \mathcal{T}_n$  there exists a  $\hat{T} \in \mathcal{T}_n$  with  $w_n(\hat{T}) > 0$  which covers  $T$ .

Now we are ready to define our first master algorithm, given in Algorithm 1. We note that the consistency of  $\{w_t\}$  implies that, for any time horizon  $n$ , Algorithm 1 is equivalent to the exponentially weighted average forecaster (1) with the set of experts  $\{(\mathcal{A}, T) : T \in \mathcal{T}_n, w_n(T) > 0\}$  and initial weights  $w_n(T)$  for  $(\mathcal{A}, T)$ . The performance and the computational complexity of the algorithm heavily depend on the properties of  $w_t$ ; in this paper we will concentrate on judicious choices of  $w_t$  that allow efficient computation of the summations in Algorithm 1 and have good prediction performance.

---

**Algorithm 1** General tracking algorithm.

---

**Input:** prediction algorithm  $\mathcal{A}$ , weight functions  $\{w_t; t = 1, \dots, n\}$ , learning parameters  $\eta_t > 0, t = 1, \dots, n$ .  
For  $t = 1, \dots, n$  predict

$$\hat{p}_t = \frac{\sum_{T \in \mathcal{T}_t} w_t(T) e^{-\eta_t L_{t-1}(\mathcal{A}, T_{t-1})} f_{\mathcal{A}, t}(\tau_t(T))}{\sum_{T \in \mathcal{T}_t} w_t(T) e^{-\eta_t L_{t-1}(\mathcal{A}, T_{t-1})}}.$$


---

The next lemma gives an upper bound on the performance of Algorithm 1.

*Lemma 1:* Suppose  $\eta_{t+1} \leq \eta_t$  for all  $t = 1, \dots, n-1$ , the transition path  $T_n$  is covered by  $\hat{T}_n = (\hat{t}_1, \dots, \hat{t}_{C(\hat{T}_n)})$  such that  $w_n(\hat{T}_n) > 0$ , and  $\mathcal{A}$  satisfies the regret bound (5). Assume that the loss function  $\ell$  is convex in its first argument and takes values in the interval  $[0, 1]$ . Then for any meta expert  $(T_n, a)$ , the regret of Algorithm 1 is bounded as

$$\begin{aligned} \hat{L}_n - L_n(T_n, a) &\leq \sum_{c=0}^{C(\hat{T}_n)} \rho_{\mathcal{E}}(\hat{t}_{c+1} - \hat{t}_c) + \sum_{t=1}^n \frac{\eta_t}{8} + \frac{1}{\eta_n} \ln \frac{1}{w_n(\hat{T}_n)} \\ &\leq (C(\hat{T}_n) + 1) \rho_{\mathcal{E}} \left( \frac{n}{C(\hat{T}_n) + 1} \right) \\ &\quad + \sum_{t=1}^n \frac{\eta_t}{8} + \frac{1}{\eta_n} \ln \frac{1}{w_n(\hat{T}_n)}. \end{aligned} \quad (7)$$

On the other hand, if  $\ell$  is exp-concave for the value of  $\eta$  and Algorithm 1 is used with  $\eta_t \equiv \eta$ , then

$$\hat{L}_n - L_n(T_n, a)$$

$$\begin{aligned} &\leq \sum_{c=0}^{C(\hat{T}_n)} \rho_{\mathcal{E}}(\hat{t}_{c+1} - \hat{t}_c) + \frac{1}{\eta} \ln \frac{1}{w_n(\hat{T}_n)} \\ &\leq (C(\hat{T}_n) + 1) \rho_{\mathcal{E}} \left( \frac{n}{C(\hat{T}_n) + 1} \right) + \frac{1}{\eta} \ln \frac{1}{w_n(\hat{T}_n)}. \end{aligned} \quad (8)$$

*Proof:* Let  $\hat{a} = (\hat{a}_0, \dots, \hat{a}_C)$  be the expert vector such that the meta experts  $(T, a)$  and  $(\hat{T}, \hat{a})$  perform identically. Then clearly

$$\begin{aligned} \hat{L}_n - L_n(T, a) &= \hat{L}_n - L_n(\mathcal{A}, \hat{T}_n) + L_n(\mathcal{A}, \hat{T}_n) - L_n(\hat{T}_n, \hat{a}). \end{aligned}$$

Using (5) and the concavity of  $\rho_{\mathcal{E}}$ , we get

$$\begin{aligned} &L_n(\mathcal{A}, \hat{T}_n) - L_n(\hat{T}_n, \hat{a}) \\ &= \sum_{c=0}^{C(\hat{T}_n)} \left( L_{\mathcal{A}}(\hat{t}_c, \hat{t}_{c+1}) - L_{\hat{a}_c}(\hat{t}_c, \hat{t}_{c+1}) \right) \\ &\leq \sum_{c=0}^{C(\hat{T}_n)} \rho_{\mathcal{E}}(\hat{t}_{c+1} - \hat{t}_c) \\ &\leq (C(\hat{T}_n) + 1) \rho_{\mathcal{E}} \left( \frac{n}{C(\hat{T}_n) + 1} \right). \end{aligned} \quad (9)$$

Assume that the loss function  $\ell$  is convex in its first argument and takes values in the interval  $[0, 1]$ . Since Algorithm 1 is equivalent to the exponentially weighted average forecaster with experts  $\{(\mathcal{A}, T) : T \in \mathcal{T}_n, w_n(T) > 0\}$  and initial weights  $w_n(T)$ , we can apply the bound (2) to obtain

$$\hat{L}_n \leq L_n(\mathcal{A}, \hat{T}_n) + \frac{1}{\eta} \ln \frac{1}{w_n(\hat{T}_n)} + \sum_{t=1}^n \frac{\eta_t}{8}.$$

Combining this with (9) proves (7).

Now assume  $\ell$  is exp-concave. Then by [4, Lemma 1],

$$\hat{L}_n - L_n(\mathcal{A}, \hat{T}_n) \leq \frac{1}{\eta} \ln \frac{1}{w_n(\hat{T}_n)}. \quad (10)$$

This, together with (9), implies (8). ■

### B. The weight function

One may interpret the weight function  $\{w_t\}$  as the conditional probability that a new segment is started, given the beginning of the current segment and the current time instant. In this case one may define  $\{w_t\}$  in terms of a time-inhomogeneous Markov chain  $\{U_t; t = 1, 2, \dots\}$  whose state space at time  $t$  is  $\{1, \dots, t\}$ . Starting from state  $U_1 = 1$ , at any time instant  $t$ , the Markov-chain either stays where it was at time  $t-1$  or switches to state  $t$ . The distribution of  $\{U_t\}$  is uniquely determined by prescribing  $\mathbb{P}(U_1 = 1) = 1$  and for  $1 \leq t' < t$ ,

$$\begin{aligned} \mathbb{P}(U_t = t | U_{t-1} = t') &= 1 - \mathbb{P}(U_t = t' | U_{t-1} = t') = p(t|t') \end{aligned} \quad (11)$$

where the so-called *switch probabilities*  $p(t|t')$  need only satisfy  $p(t|t') \in [0, 1]$  for all  $1 \leq t' < t$ . A realization of this Markov chain uniquely determines a transition path:

$T_t(u_1, \dots, u_t) = (t_1, \dots, t_C) \in \mathcal{T}_t$  if and only if  $u_{k-1} \neq u_k$  for  $k \in \{t_1, \dots, t_C\}$ , and  $u_{k-1} = u_k$  for  $k \notin \{t_1, \dots, t_C\}$ ,  $2 \leq k \leq t$ . Inverting this correspondence, any  $T \in \mathcal{T}_t$  uniquely determines a realization  $(u_1, \dots, u_t)$ . Now the weight function is given for all  $t \geq 1$  and  $T \in \mathcal{T}_t$  by

$$w_t(T) = \mathbb{P}(U_1 = u_1, \dots, U_t = u_t) \quad (12)$$

where  $(u_1, \dots, u_t)$  is such that  $T = T(u_1, \dots, u_t)$ . It is easy to check that  $\{w_t\}$  satisfies the two conditions in (6). Clearly, the switch probabilities  $p(t|t')$  uniquely determine  $\{w_t\}$ . The above structural assumption on  $\{w_t\}$ , originally introduced in [2], greatly reduces the possible ways of weighting different transition paths, allowing implementation of Algorithm 1 with complexity at most  $O(n^2)$  (if one step of  $\mathcal{A}$  can be implemented in constant time), instead of the potentially exponential time complexity of the algorithm in the naive implementation; see Section III-C.

Some examples that have been proposed for this construction (given in terms of the switch probabilities) include

- $w^{HW}$ , used in [4], is defined by  $p_{HW}(t|t') = \alpha$  for some  $0 < \alpha < 1$ .
- $w^{HS}$ , used in [6], [8], [11], is defined by  $p^{HS}(t|t') = 1/t$ .
- $w^{KT}$ , used in [2], is defined by

$$p_{KT}(t|t') = \frac{1/2}{t - t' + 1} \quad (13)$$

which is the Krichevsky-Trofimov estimate [13] for binary sequences of the probability that after observing an all zero sequence of length  $t - t'$ , the next symbol will be a one. Using standard bounds on the Krichevsky-Trofimov estimate, it is easy to show (see, e.g., [2]) that for any  $T \in \mathcal{T}_n$  with segment lengths  $s_0, s_1, \dots, s_C \geq 1$  (satisfying  $\sum_{c=0}^C s_c = n$ )

$$\ln \frac{1}{w^{KT}(T)} \leq \frac{1}{2} \sum_{c=0}^C \ln s_c + (C+1) \ln 2. \quad (14)$$

- $w^{\mathcal{L}^1}$  and  $w^{\mathcal{L}^2}$  used in [3] (similar weight functions were considered in [5]), are defined as follows: for a given  $0 < \epsilon < 1$ ,<sup>2</sup> let  $\pi_j = 1/j^{1+\epsilon}$ ,  $Z_t = \sum_{j=1}^t \pi(j)$  (with  $Z_0 = 0$  and  $Z_\infty = \sum_{j=1}^\infty \pi(j)$ ). Then  $w^{\mathcal{L}^1}$  and  $w^{\mathcal{L}^2}$  are defined, respectively, by

$$p_{\mathcal{L}^1}(t|t') = \frac{\pi(t-1)}{(Z_\infty - Z_{t-2})}$$

and

$$p_{\mathcal{L}^2}(t|t') = \frac{\pi(t-t')}{(Z_\infty - Z_{t-t'+1})}.$$

Here we consider the weights  $w^{\mathcal{L}^1}$ . It is shown in [3, proof of Eq. (39)] that for any  $T \in \mathcal{T}_n$ ,

$$\ln \frac{1}{w_n^{\mathcal{L}^1}(T)} \leq (C_n(T) + \epsilon) \ln n + \ln(1 + \epsilon) - C_n(T) \ln \epsilon. \quad (15)$$

<sup>2</sup>The upper bound  $\epsilon < 1$  is missing from [3], although it is implicitly required in the proof.

### C. A low-complexity algorithm

Efficient implementation of Algorithm 1 hinges on three factors: (i) Algorithm  $\mathcal{A}$  can be efficiently implemented; (ii) the exponential weighting step can be efficiently implemented; which is facilitated by (iii) the availability of the losses  $L_{\mathcal{A}}(t', t)$  at each time instant  $t$  for all  $1 \leq t' \leq t$  in the sense that these losses can be computed efficiently. In what follows we assume that (i) and (iii) hold and develop a method for (ii) via constructing a new weight function  $\{\hat{w}_t\}$  that significantly reduces the complexity of implementing Algorithm 1.

First, we observe that the predictor  $\hat{p}_t$  of Algorithm 1 can be rewritten as

$$\hat{p}_t = \frac{\sum_{t'=1}^t v_t(t') f_{\mathcal{A},t}(t')}{\sum_{t'=1}^t v_t(t')} \quad (16)$$

where the weights  $v_t$  are given by

$$v_t(t') = \sum_{T \in \mathcal{T}_t: \tau_t(T)=t'} w_t(T) e^{-\eta_t L_{t-1}(\mathcal{A}, T_{t-1})}. \quad (17)$$

Note that  $v_t(t')$  gives the weighted sum of the exponential weights of all transition paths with the last switch at  $t'$ .

If the learning parameters  $\eta_t$  are constant during the time horizon, the above means that Algorithm 1 can be implemented efficiently by keeping a weight  $v_t(t')$  at each time instant  $t$  for every possible starting point of a segment  $t' = 1, \dots, t$ . Indeed, if  $\eta_t = \eta$  for all  $t$ , then (17), (11), and (12) imply that each  $v_t(t')$  can be computed recursively in  $O(t)$  time from the  $v_{t-1}$  (setting  $v_1(1) := 1$  at the beginning) using the switch probabilities defining  $w_t$  as follows:

$$v_t(t') = \begin{cases} v_{t-1}(t')(1 - p(t|t'))e^{-\eta \ell(f_{\mathcal{A},t-1}(t'), y_{t-1})} & \text{for } t' = 1, \dots, t-1, \\ \sum_{t''=1}^{t-1} v_{t-1}(t'')p(t|t'')e^{-\eta \ell(f_{\mathcal{A},t-1}(t''), y_{t-1})} & \text{for } t' = t. \end{cases} \quad (18)$$

Using this recursion, the overall complexity of computing the weights during  $n$  rounds is  $O(n^2)$ . Furthermore, (16) means that one needs to start an instance of  $\mathcal{A}$  for each possible starting point of a segment. If the complexity of running algorithm  $\mathcal{A}$  for  $n$  time steps is  $O(n)$  (i.e., computing  $\mathcal{A}$  at each time instant has complexity  $O(1)$ ), then the overall complexity of our algorithm becomes  $O(n^2)$ .

It is clearly not a desirable feature that the amount of computation per time round grows (linearly) with the horizon  $n$ . While we don't know how to completely eliminate this ever-growing computational demand, we are able to moderate this growth significantly. To this end, we modify the weight functions in such a way that at any time instant  $t$  we allow at most  $O(g \ln t)$  actual segments with positive probability (i.e., segments containing  $t$  that belong to sample paths with positive weights), where  $g > 0$  is a parameter of the algorithm (note that  $g$  may depend on, e.g., the time horizon  $n$ ). Specifically, we will construct a new weight function  $\hat{w}_t$  such that

$$|\{\tau_t(T) : \hat{w}_t(T_t) > 0, T \in \mathcal{T}_n\}| \leq \left\lceil \frac{g}{2} \right\rceil (\lceil \log t \rceil + 1)$$

where  $\log$  denotes base-2 logarithm. By doing so, the time and space complexity of the algorithm becomes  $O(g \ln n)$  times

more than that of algorithm  $\mathcal{A}$ , as we need to run  $O(g \ln n)$  instances of  $\mathcal{A}$  in parallel and the number of non-zero terms in (18) and (16) is also  $O(g \ln n)$  (here we exclude the trivial case where  $\mathcal{A}$  has zero space complexity; also note that the time-complexity of  $\mathcal{A}$  is at least linear in  $n$  since it has to make a prediction at each time instant). Thus, in case of a linear-time-complexity algorithm  $\mathcal{A}$ , the overall complexity of Algorithm 1 becomes  $O(gn \ln n)$ .

In order to construct the new weight function, at each time instant  $t$  we force some segments to end. Then any path that contains such a segment will start a new segment at time  $t$  (and hence the corresponding vector of transitions contains  $t$ ). Specifically, any time instant  $s$  can be uniquely written as  $o2^u$  with  $o$  being a positive odd number and  $u$  a nonnegative integer (i.e.,  $2^u$  is the largest power of 2 that divides  $s$ ). We specify that a segment starting at  $s$  can “live” for at most  $g2^u$  time instants, where  $g > 0$  is a parameter of the algorithm, so that at time  $s + g2^u$  we force a switch in the path. More precisely, given any switch probability  $p(t|t')$  for all  $t' < t$ , we define a new switch probability

$$\hat{p}(t|t') = 1 - h_t(t')(1 - p(t|t')) \quad (19)$$

where

$$h_t(s) = \begin{cases} 1 & \text{if } s \leq t < s + g2^u, \\ 0 & \text{otherwise.} \end{cases}$$

Thus  $h_t(s) = 1$  if and only if a segment started at  $s$  is still valid at time  $t$ . In terms of the Markov chain  $\{U_t\}$  introduced in (11), the new switch probabilities in definition (19) mean that if the chain is in state  $t'$  at time  $t-1$  such that  $h_t(t') = 1$ , then the chain switches to state  $t$  with the original switch probability  $p(t|t')$  and remains at state  $t'$  with probability  $1 - p(t|t')$ ; but if  $h_t(t') = 0$ , then the chain switches to state  $t$  with probability 1. In this way, given the switch probabilities  $p(t|t')$  and the associated weight function  $\{w_t\}$ , we can define a new weight function  $\{\hat{w}_t\}$  via the new switch probabilities  $\hat{p}(t|t')$  and the procedure described in Section III-B. Note that the definition of  $\{\hat{w}_t\}$  implies that for a transition path  $T \in \mathcal{T}_t$  either

$$\hat{w}_t(T) = 0 \quad \text{or} \quad \hat{w}_t(T) \geq w_t(T). \quad (20)$$

The above procedure is a common generalization of several algorithms previously reported in the literature for pruning the transition paths. Specifically,  $g = 1$  yields the procedure of [9],  $g = 3$  yields our previous procedure [10],  $g = 4$  yields the method of [11], while  $g = n$  yields the original weighting  $\{w_t\}$  without pruning. We will show that the time complexity of the method with a constant  $g$  (i.e., when  $g$  is independent of the time horizon  $n$ ) is, in each time instant, at most  $O(\ln n)$  times the complexity of one step of  $\mathcal{A}$ , while the time complexity of the algorithm without pruning is  $O(n)$  times the complexity of  $\mathcal{A}$ . Complexities that interpolate between these two extremes can be achieved by setting  $g = o(n)$  appropriately.

We say that a segment at time instant  $t$  is *alive* if it contains  $t$  and is *valid* if there is a path  $T_t$  with  $\hat{w}_t(T_t) > 0$  that contains exactly that segment. In what follows we assume that the original switch probabilities  $p(t|t')$  associated with the  $w_t$

satisfy  $p(t|t') \in (0, 1)$  for all  $1 \leq t' < t$ . (Note that the weight function examples introduced in Section III-B all satisfy this condition.) The condition implies that  $w_t(T_t) > 0$  for all  $T_t \in \mathcal{T}_t$ . Furthermore, if  $T_t = (t_1, \dots, t_C) \in \mathcal{T}_t$  satisfies  $t_{i+1} - t_i < g2^{u_{t_i}}$ ,  $i = 1, \dots, C$ , where  $u_{t_i}$  is the largest power of 2 divisor of  $t_i$ , then from (19) we get  $\hat{w}_t(T) > 0$ .

The next lemma gives a characterization of when  $h_t(s) = 1$  and, as a consequence, bounds the number of valid segments that are alive at  $t$ .

**Lemma 2:** Let  $t = \sum_{i=1}^m 2^{u_i}$  be the binary form of  $t$  with  $0 \leq u_1 < u_2 < \dots < u_m$ ,  $s_k = \sum_{i=k}^m 2^{u_i}$ , and  $u_0 = -1$ . Then  $h_t(s) = 1$  if and only if  $s = s_k - j2^u$  for some  $u_{k-1} < u \leq u_k$  and  $j \in \{0, \dots, g-1\}$  such that  $2^u$  is the largest 2-power divisor of  $s$ ; in particular,  $j$  is even if  $u = u_k$  for some  $k \in \{1, \dots, m\}$ , and odd otherwise. As a consequence, at any time instant  $t$  there are at most  $\lceil g/2 \rceil (\lceil \log t \rceil + 1)$  segments that are valid and alive.

*Proof:* It is clear that for any  $s$  satisfying the conditions of the lemma,  $h_t(s) = 1$  since  $s + g2^u = s_k - j2^u + g2^u \geq s_k + 2^u > t \geq s$ . To prove the other direction, consider an  $s \in [1, t]$ ; assume  $h_t(s) = 1$  and denote the largest 2-power divisor of  $s$  by  $2^u$ . By definition,  $h_t(s) = 1$  if and only if  $s + j2^u \leq t < s + (j+1)2^u$  for some  $j \in \{0, \dots, g-1\}$ . After reordering we obtain

$$t - (j+1)2^u < s \leq t - j2^u. \quad (21)$$

Let  $k \in \{1, \dots, m\}$  be the unique index such that  $u_{k-1} < u \leq u_k$  (note that  $u \leq u_m$  always holds). Then  $2^u$  divides  $s_k$ , and  $s_k \leq t < s_k + 2^u$ . Combining this inequality with (21) gives  $s_k - (j+1)2^u < s < s_k - (j-1)2^u$ . Taking into account that both  $s$  and  $s_k$  are divisible by  $2^u$ , we obtain  $s = s_k - j2^u$ . Furthermore, since  $2^u$  is the largest 2-power divisor of  $s$ ,  $j$  must be even when  $u = u_k$  for some  $k \in \{1, \dots, m\}$ , and odd otherwise.

Finally, for any  $u \in \{0, 1, \dots, u_m\}$ , the set

$$\{s = s_k - j2^u : u_{k-1} < u \leq u_k, j = 0, \dots, g-1, \\ 2^u \text{ is the largest 2-power divisor of } s\}$$

has at most  $\lceil g/2 \rceil$  elements. Since  $u_m = \lfloor \log t \rfloor$ , the proof is complete. ■

Note that for  $g = 1$  the valid segments that are alive at  $t$  start exactly at  $s_k$ ,  $k = 1, \dots, m$ , and so the number of valid segments at time  $t$  is exactly the number of 1's in the binary form of  $t$  [9]. The above lemma implies that Algorithm 1 can be implemented efficiently with the proposed weight function  $\{\hat{w}_t\}$ .

**Theorem 1:** Assume Algorithm 1 is run with weight function  $\{\hat{w}_t\}$  derived using any  $g > 0$  from any weight function  $\{w_t\}$  defined as in Section III-B. If  $\eta_t = \eta$  for some  $\eta > 0$  and all  $t = 1, \dots, n$ , then the time and space complexity of Algorithm 1 is  $O(g \ln n)$  times the time and space complexity of  $\mathcal{A}$ , respectively.

*Proof:* The result follows since Lemma 2 implies that the number of non-zero terms in (18) and (16) is always  $O(g \ln t)$ . ■

### D. Regret bounds

To bound the regret, we need the following lemma which shows that any segment  $[t, t']$  can be covered with at most  $\left\lceil \frac{\log(t'-t)}{\log(g+1)} \right\rceil + 1$  valid segments.

**Lemma 3:** For any  $T \in T_n$ , there exists  $\hat{T} \in T_n$  such that for any segment  $[t, t']$  of  $T$  with  $1 \leq t < t' \leq n+1$ ,

- (i)  $\hat{w}_{t'}(\hat{T}) > 0$ ,  $t$  and  $t'$  are switch points of  $\hat{T}$  (if  $t' = n+1$ , it is considered as a switch point), and  $\hat{T}$  contains at most  $l = \left\lceil \frac{\log(t'-t)}{\log(g+1)} \right\rceil + 1$  segments in  $[t, t']$ ;
- (ii) if the switch points of  $\hat{T}$  in  $[t, t']$  are  $t_1 := t < t_2 < \dots < t_{l'} < t_{l'+1} := t'$ , then  $l' \leq l$ , and for any nondecreasing function  $f : [0, \infty) \rightarrow [0, \infty)$ ,

$$\begin{aligned} & \sum_{i=1}^{l'} f(t_{i+1} - t_i) \\ & \leq \sum_{i=0}^{l'-2} f\left(\frac{t' - t}{2^{i \lfloor \log(g+1) \rfloor}}\right) + f(t' - t) \quad (22) \\ & \leq \int_0^{\frac{\log(t'-t)}{\lfloor \log(g+1) \rfloor}} f\left(\frac{t' - t}{2^{x \lfloor \log(g+1) \rfloor}}\right) dx + 2f(t' - t) \quad (23) \end{aligned}$$

where the second summation in (22) is empty if  $l' = 1$ .

*Remark:* Note that it is possible to obtain for  $l$  the less compact and harder-to-handle formula

$$l = \left\lceil \frac{\log \frac{t' - t + \frac{1}{2^{\lfloor \log(g+1) \rfloor - 1}}}{2^{\lfloor \log(g+1) \rfloor - 1} - 1 + \frac{1}{2^{\lfloor \log(g+1) \rfloor - 1}}}}{\lfloor \log(g+1) \rfloor} \right\rceil + 1$$

by taking into account that the last segment  $[t_l, t_{l+1}]$  in the construction of the proof can always be defined to be of length at least  $\lfloor \log(g+1) \rfloor 2^{u_l}$ . Furthermore, for  $g = 1$  it follows from [9] that the last term is not needed in (22), and hence the latter bound can be strengthened to

$$\sum_{i=1}^{l'} f(t_{i+1} - t_i) \leq \sum_{i=0}^{\lfloor \log(t'-t) \rfloor} f(2^i). \quad (24)$$

*Proof:* Clearly, it is enough to define  $\hat{T}$  independently in each segment  $[t, t']$  of  $T$ . We construct the switch points  $t_1 < t_2 < \dots < t_{l'}$  of  $\hat{T}$  in this interval, for some  $l' \leq l$ , and an auxiliary variable  $t_{l'+1} \geq t'$  one by one such that  $t_1 = t$ ,  $t_{l'} < t'$  and, defining  $u_j$  as the largest 2-power divisor of  $t_j$ ,

$$u_{j+1} - u_j \geq \lfloor \log(g+1) \rfloor \quad (25)$$

for  $j = 1, \dots, l' - 1$ . Assume that we have already defined  $t_1, \dots, t_i$  satisfying (25) for  $j = 1, \dots, i - 1$ . Then a segment starting at  $t_i$  may be alive with positive probability at any time instant in  $[t_i, t_i + g2^{u_i}]$ . Define  $u_{i+1}$  to be the largest nonnegative integer such that there is an  $s \in [t_i + 1, t_i + g2^{u_i}]$  such that  $2^{u_{i+1}}$  divides  $s$ . Then  $s2^{-u_i}$  belongs to the set  $\mathcal{S}_i = \{t_i 2^{-u_i}, t_i 2^{-u_i} + 1, t_i 2^{-u_i} + 2, \dots, t_i 2^{-u_i} + g\}$  (although, clearly,  $s2^{-u_i} \neq t_i 2^{-u_i}$ ). Since  $\mathcal{S}_i$  is a set of  $g+1$  consecutive integers, it has an element  $a$  that is divisible by  $2^{\lfloor \log(g+1) \rfloor}$ , and this element is not the odd number  $t_i 2^{-u_i}$ . Thus  $2^{u_i}a \in$

$[t_i + 1, t_i + g2^{u_i}]$  and since  $2^{u_i}a$  is divisible by  $2^{u_i + \lfloor \log(g+1) \rfloor}$ , the maximal property of the 2-power divisor  $2^{u_{i+1}}$  of  $s$  implies that  $u_{i+1} \geq u_i + \lfloor \log(g+1) \rfloor$ . Therefore, defining  $t_{i+1} = s$ , its largest 2-power divisor is  $2^{u_{i+1}}$ , proving (25) for  $j = i$  (note that it is easy to show that the choice of  $s$ , and hence that of  $t_{i+1}$ , is unique).

Now let  $l'$  be the smallest integer such that  $t_{l'+1} \geq t'$ . To prove part (i) of the lemma, it is sufficient to show that  $l' \leq l$  and the segments  $[t_1, t_2], [t_2, t_3], \dots, [t_{l'-1}, t_{l'}], [t_{l'}, t']$  cover  $[t, t']$ , which is clearly true if  $t_{l'+1} \geq t'$ . From (25) and the fact that  $t_{i+1} - t_i$  is divisible by  $2^{u_i}$ , we have

$$\begin{aligned} t_{l'+1} & \geq t + \sum_{i=1}^l 2^{u_i} = t + \sum_{i=1}^l 2^{u_1 + \sum_{j=2}^i (u_j - u_{j-1})} \\ & \geq t + \sum_{i=1}^l 2^{u_1 + \sum_{j=2}^i \lfloor \log(g+1) \rfloor} \\ & = t + \sum_{i=0}^{l-1} 2^{u_1 + i \lfloor \log(g+1) \rfloor} \\ & = t + 2^{u_1} \frac{2^{l \lfloor \log(g+1) \rfloor} - 1}{2^{\lfloor \log(g+1) \rfloor} - 1} \\ & \geq t + 2^{(l-1) \lfloor \log(g+1) \rfloor} \geq t' \end{aligned}$$

where in the last step we used the definition of  $l$ . This finishes the proof of (i).

To prove (ii), we first show that the transition path  $\hat{T}$  constructed above satisfies (22), where, with a slight abuse of notation, we redefine  $t_{l'+1}$  from part (i) to be  $t'$ . First notice that since  $t + g2^{u_{l'-1}} \leq t_{l'} + g2^{u_{l'-1}} < t'$ , we have  $u_{l'-1} \leq \left\lfloor \log \frac{t' - t}{g} \right\rfloor$ . Repeated application of (25) implies, for any  $i = 1, \dots, l' - 1$ ,

$$u_i \leq \left\lfloor \log \frac{t' - t}{g} \right\rfloor - (l' - 1 - i) \lfloor \log(g+1) \rfloor$$

and

$$\begin{aligned} t_{i+1} - t_i & \leq g2^{\left\lfloor \log \frac{t' - t}{g} \right\rfloor - (l' - 1 - i) \lfloor \log(g+1) \rfloor} \\ & \leq g2^{\log \frac{t' - t}{g} - (l' - 1 - i) \lfloor \log(g+1) \rfloor} \\ & = (t' - t)2^{-(l' - 1 - i) \lfloor \log(g+1) \rfloor}. \end{aligned}$$

Using the crude estimate  $t' - t_l \leq t' - t$  finishes the proof of (22). The last inequality (23) holds trivially for  $l = 1$ , and holds for  $l \geq 2$  since

$$\begin{aligned} & \sum_{i=0}^{l'-2} f\left(\frac{t' - t}{2^{i \lfloor \log(g+1) \rfloor}}\right) \\ & = f(t' - t) + \sum_{i=1}^{l'-2} f\left(\frac{t' - t}{2^{i \lfloor \log(g+1) \rfloor}}\right) \\ & \leq f(t' - t) + \int_0^{\left\lceil \frac{\log(t'-t)}{\lfloor \log(g+1) \rfloor} \right\rceil - 1} f\left(\frac{t' - t}{2^{x \lfloor \log(g+1) \rfloor}}\right) dx \\ & \leq f(t' - t) + \int_0^{\frac{\log(t'-t)}{\lfloor \log(g+1) \rfloor}} f\left(\frac{t' - t}{2^{x \lfloor \log(g+1) \rfloor}}\right) dx. \end{aligned}$$

■



Taking into account that  $C(T_n) \leq C(\hat{T}_n)$  if  $\hat{T}_n$  covers  $T_n$ , Lemma 3 trivially implies the following bounds.

*Lemma 4:* For any  $T_n \in \mathcal{T}_n$  there exists a  $\hat{T}_n \in \mathcal{T}_n$  with  $\hat{w}_n(\hat{T}_n) > 0$  such that  $\hat{T}_n$  covers  $T_n$  and

$$C(T_n) \leq C(\hat{T}_n) \leq (C(T_n) + 1)L_{C(T_n),n} - 1 \quad (26)$$

where

$$L_{C,n} = \begin{cases} \left\lceil \frac{\log n}{\log(g+1)} \right\rceil + 1 & \text{if } C = 0, \\ \left\lceil \frac{\log \frac{n}{C+1}}{\log(g+1)} \right\rceil + 2 & \text{if } C \geq 1. \end{cases} \quad (27)$$

*Proof:* The lower bound is trivial, and the upper bound directly follows from Lemma 3 for  $C(T_n) = 0$ . For  $C(T_n) \geq 1$  the upper bounds follow since on each segment of  $T_n$  we can define  $\hat{T}_n$  as in the proof of Lemma 3. Hence, if  $T = (t_1, \dots, t_C; n)$ , then

$$\begin{aligned} C(\hat{T}_n) + 1 &\leq \sum_{i=1}^{C+1} \left( \left\lceil \frac{\log(t_i - t_{i-1})}{\log(g+1)} \right\rceil + 1 \right) \\ &\leq \sum_{i=1}^{C+1} \left( \frac{\log(t_i - t_{i-1})}{\log(g+1)} + 2 \right) \\ &\leq (C+1) \left( \frac{\log \frac{n}{C+1}}{\log(g+1)} + 2 \right) \end{aligned}$$

where in the last step we used Jensen's inequality and the concavity of the logarithm. ■

We now apply the preceding construction and results to the weight function  $\{w_t\} = \{w_t^{\mathcal{L}^1}\}$  to obtain our main theorem:

*Theorem 2:* Assume Algorithm 1 is run with  $g > 0$  and weight function  $\{\hat{w}_t^{\mathcal{L}^1}\}$  for some  $0 < \epsilon < 1$  (derived from  $\{w_t^{\mathcal{L}^1}\}$ ), based on a prediction algorithm that satisfies (5) for some  $\rho_{\mathcal{E}}$ . Let  $L_{C,n}$  be defined by (27). If  $\ell$  is convex in its first argument and takes values in the interval  $[0, 1]$  and  $\eta_{t+1} \leq \eta_t$  for  $t = 1, \dots, n-1$ , then for all  $n \geq 1$  and any  $T \in \mathcal{T}_n$ , the tracking regret satisfies

$$\begin{aligned} \hat{L}_n - L_n(T, a) &\leq L_{C(T),n}(C(T) + 1)\rho_{\mathcal{E}} \left( \frac{n}{L_{C(T),n}(C(T) + 1)} \right) \\ &\quad + \sum_{t=1}^n \frac{\eta_t}{8} + \frac{r_n(L_{C(T),n}(C(T) + 1) - 1)}{\eta_n} \end{aligned} \quad (28)$$

where the function  $r_n(C)$  is defined as

$$r_n(C) = (C + \epsilon) \ln n + \ln(1 + \epsilon) - C \ln \epsilon.$$

Furthermore, for  $\epsilon \leq 1/2$  and  $n \geq 5$ , the adaptive regret of the algorithm satisfies

$$R_n^a \leq L_{0,n}\rho_{\mathcal{E}} \left( \frac{n}{L_{0,n}} \right) + \sum_{t=1}^n \frac{\eta_t}{8} + \frac{r'_n(L_{0,n} - 1)}{\eta_n} \quad (29)$$

where the function  $r'_n(C)$  is defined as

$$r'_n(C) = (C + 1) \ln n - (C + 1) \ln \epsilon.$$

On the other hand, if  $\ell$  is exp-concave for some  $\eta > 0$  and we let  $\eta_t = \eta$  for  $t = 1, \dots, n$  in Algorithm 1, then for any  $n \geq 1$  and  $T \in \mathcal{T}_n$  the tracking regret satisfies

$$\begin{aligned} \hat{L}_n - L_n(T, a) &\leq L_{C(T),n}(C(T) + 1)\rho_{\mathcal{E}} \left( \frac{n}{L_{C(T),n}(C(T) + 1)} \right) \\ &\quad + \frac{r_n(L_{C(T),n}(C(T) + 1) - 1)}{\eta} \end{aligned} \quad (30)$$

while for  $0 < \epsilon \leq 1/2$  and  $n \geq 5$ , the adaptive regret can be bounded as

$$R_n^a \leq L_{0,n}\rho_{\mathcal{E}} \left( \frac{n}{L_{0,n}} \right) + \frac{r'_n(L_{0,n} - 1)}{\eta}. \quad (31)$$

*Proof:* First we show the bounds for the tracking regret. To prove the theorem, let  $\hat{T}_n$  be defined as in Lemma 1, and we bound the first and last terms on the right-hand side of (7) and (8) (with  $\hat{w}_n^{\mathcal{L}^1}$  in place of  $w_n$ ). Note that the conditions on  $\rho_{\mathcal{E}}$  imply that  $x\rho_{\mathcal{E}}(y/x)$  is a nondecreasing function of  $x$  for any fixed  $y > 0$  (this follows since  $\rho_{\mathcal{E}}(z)/z = (\rho_{\mathcal{E}}(z) - 0)/(z - 0)$  is a nonincreasing function of  $z > 0$  by the concavity of  $\rho_{\mathcal{E}}$ , and hence  $z\rho_{\mathcal{E}}(1/z)$  is nondecreasing). Combining this with the bounds on  $C(T_n)$  in Lemma 4 implies

$$\begin{aligned} (C(\hat{T}_n) + 1)\rho_{\mathcal{E}} \left( \frac{n}{C(\hat{T}_n) + 1} \right) &\leq L_{C(T),n}(C(T) + 1)\rho_{\mathcal{E}} \left( \frac{n}{L_{C(T),n}(C(T) + 1)} \right). \end{aligned}$$

The last term  $(1/\eta_n) \ln(1/\hat{w}_n^{\mathcal{L}^1}(\hat{T}_n))$  in (7) and (8) can be bounded by noting that  $1/\hat{w}_n^{\mathcal{L}^1}(\hat{T}_n) \leq 1/w_n^{\mathcal{L}^1}(\hat{T}_n)$  by (20) and the latter can be bounded using (15); this is given by  $r_n$ . This finishes the proof of the tracking regret bounds.

Next we prove the bounds for the adaptive regret. Assume we want to bound the regret of our algorithm in a segment  $[t, t']$  with  $1 \leq t < t' \leq n + 1$ . By Lemma 3 there exists a transition path  $\hat{T}_{t'-1}$  such that it has a switch point at  $t$ , has at most  $l = \left\lceil \frac{\log(t' - t)}{\log(g+1)} \right\rceil + 1 \leq L_{0,n}$  segments in  $[t, t']$ , and  $\hat{w}_n(\hat{T}_n) > 0$ . Let  $\hat{t}_1, \hat{t}_2, \dots, \hat{t}_{\hat{C}}$  denote the switch points of  $\hat{T}_n$  in  $[t + 1, t']$  where  $\hat{C} < l$ , and let  $\hat{t}_0 = t$  and  $\hat{t}_{\hat{C}+1} = t'$ . Notice that, since we are interested in the performance of the algorithm only in the interval  $[t, t']$ , a modified version of Lemma 1 can be applied, where the loss is considered only in the interval  $[t, t']$  and the weight of  $\hat{T}_n$  can be thought to be the sum of the weight of all transition paths that agree with  $\hat{T}_n$  in  $[t, t']$ . Specifically, letting  $\mathcal{T}_{t,t'}(\hat{T}_{t'-1}) = \{T \in \mathcal{T}_{t'-1} : T \text{ and } \hat{T}_{t'-1} \text{ agree on } [t, t']\}$  and  $\hat{w}_{t,t'}^{\mathcal{L}^1}(\hat{T}_n) = \sum_{T \in \mathcal{T}_{t,t'}} \hat{w}_{t'-1}^{\mathcal{L}^1}(T)$ , it can be shown similarly to Lemma 1 that in the case of a loss function that is convex in its first argument and takes values in  $[0, 1]$ , for any expert  $i \in \mathcal{E}$ ,

$$\sum_{s=t}^{t'-1} (\ell(\hat{p}_s, y_s) - \ell(a, y_s))$$

$$\leq (\widehat{C} + 1)\rho_{\mathcal{E}}\left(\frac{n}{\widehat{C} + 1}\right) + \sum_{s=t}^{t'-1} \frac{\eta_s}{8} + \frac{1}{\eta_{t'-1}} \ln \frac{1}{\hat{w}_{t,t'}(\widehat{T}_{t'-1})}. \quad (32)$$

Now  $-\ln \hat{w}_{t,t'}^{\mathcal{L}_1}(\widehat{T}_{t'-1})$  can be bounded in a similar way as  $-\ln \hat{w}_n^{\mathcal{L}_1}(T_n)$  in [3]: For  $t = 1$  we can use (15). For  $t \geq 2$  it can be shown, following the proof of (15) in [3], that

$$\begin{aligned} \ln \frac{1}{\hat{w}_{t,t'}^{\mathcal{L}_1}(\widehat{T}_{t'-1})} &\leq (\widehat{C} + 1) \ln(t' - 1) - (\widehat{C} + 1) \ln \epsilon \\ &\leq (\widehat{C} + 1) \ln n - (\widehat{C} + 1) \ln \epsilon \end{aligned} \quad (33)$$

whenever  $\epsilon \leq 1/2$ . Indeed, let  $B_t$  denote the event that  $t$  is a switch point and let  $A_{t_1, \dots, t_{\widehat{C}}}$  denote the event that  $t_1, \dots, t_{\widehat{C}}$  are the switch points in  $[t+1, t']$ . Since the switch probabilities  $p_{\mathcal{L}_1}(s|s')$  are independent of  $s'$  and  $1 - p_{\mathcal{L}_1}(s|s') = \frac{Z_{\infty} - Z_{s-1}}{Z_{\infty} - Z_{s-2}}$ , for  $\epsilon \leq 1/2$ , we have

$$\begin{aligned} \hat{w}_{t,t'}^{\mathcal{L}_1}(\widehat{T}_{t'-1}) &= \mathbb{P}(B_t) \mathbb{P}(A_{t_1, \dots, t_{\widehat{C}}} | B_t) \\ &\geq \prod_{c=0}^{\widehat{C}} \frac{\pi(t_c - 1)}{Z_{\infty} - Z_{t_c - 2}} \left( \prod_{\tau=t_c+1}^{t_{c+1}-1} \frac{Z_{\infty} - Z_{\tau-1}}{Z_{\infty} - Z_{\tau-2}} \right) \\ &= \prod_{s=t}^{t'-1} \frac{Z_{\infty} - Z_{s-1}}{Z_{\infty} - Z_{s-2}} \cdot \prod_{c=0}^{\widehat{C}} \frac{\pi(t_c - 1)}{Z_{\infty} - Z_{t_c - 1}} \\ &= \frac{Z_{\infty} - Z_{t'-2}}{Z_{\infty} - Z_{t-2}} \prod_{c=0}^{\widehat{C}} \frac{\pi(t_c - 1)}{Z_{\infty} - Z_{t_c - 1}} \\ &\geq \frac{(t-1)^{1+\epsilon}}{(t'-1)^{\epsilon}(t-1+\epsilon)} \cdot \frac{\epsilon t^{1+\epsilon}}{(t+\epsilon)(t-1)^{1+\epsilon}} \cdot \frac{\epsilon^{\widehat{C}}}{(t'-1)^{\widehat{C}}} \\ &= \frac{\epsilon^{\widehat{C}+1} t^{1+\epsilon}}{(t'-1)^{\widehat{C}+\epsilon} (t-1+\epsilon)(t+\epsilon)} \\ &\geq \frac{\epsilon^{\widehat{C}+1}}{(t'-1)^{\widehat{C}+\epsilon} t^{1-\epsilon}} \geq \frac{\epsilon^{\widehat{C}+1}}{(t'-1)^{\widehat{C}+1}} \end{aligned}$$

where the second inequality follows from inequalities (36) and (38) in [3], and the third follows since  $(t-1+\epsilon)(t+\epsilon) < t^2$ .

It is easy to see that the bound in (33) is larger than (15) if  $n \geq 5$ . Thus, combining with (32) for the maximizing value  $t = 1$ ,  $t' = n + 1$  and using  $\widehat{C} \leq L_{0,n}$ , we obtain the bound (29) on the adaptive regret. A modified version of (32) (without the  $\sum_{s=t}^{t'-1} \eta_s/8$  term) yields (31)  $\blacksquare$

*Remarks:* (i) Note that the tracking regret can be trivially bounded by  $(C(T) + 1)$  times the adaptive regret (as suggested by [11]). However, the direct bounds on the tracking regret are somewhat better than this: The first term coming from the adaptive regret bound would be  $L_{0,n}(C(T) + 1)\rho_{\mathcal{E}}(n/L_{0,n})$ , which is larger than the first term  $L_{C,n}(C(T) + 1)\rho_{\mathcal{E}}(\frac{n}{L_{C,n}(C(T)+1)})$  in the tracking regret bounds. This justifies our claim for exp-concave loss functions, since the last terms will be essentially the same, although the main term in the bound is not affected. The difference is more pronounced for the case of the convex and bounded loss function, where

the middle  $\sum_t \eta_t/8$  term becomes multiplied by  $(C(T) + 1)$  if the tracking bound is computed from the adaptive regret bound, resulting in an increased constant factor in the main term.

(ii) The above theorem provides bounds on the tracking and adaptive regrets in terms of the regret bound  $\rho_{\mathcal{E}}$  of algorithm  $\mathcal{A}$ . However, in many practical situations,  $\mathcal{A}$  behaves much better than suggested by its regret bound. This behavior is also preserved in our tracking algorithms: Omitting step (9) in Lemma 1 we can replace the first term in (28) and (30) with  $L_n(\mathcal{A}, \widehat{T}_n) - L_n(\widehat{T}_n, \hat{a})$ , which is the actual regret of algorithm  $\mathcal{A}$  on the (extended) transition path  $\widehat{T}_n$ . Reordering the resulting inequality, we can see that the loss of our algorithm is not much larger than that of  $\mathcal{A}$  run on  $\widehat{T}_n$ , for example, in the exp-concave case we have

$$\widehat{L}_n - L_n(\mathcal{A}, \widehat{T}_n) \leq \frac{r_n (L_{C(T),n}(C(T) + 1) - 1)}{\eta}.$$

### E. Exponential weighting

We now apply Theorem 2 to the case where  $\mathcal{A}$  is the exponentially weighted average forecaster and the set of base experts is of size  $N$ , and discuss the obtained bounds (for simplicity we assume  $C(T) \geq 1$ , but  $C(T) = 0$  would just slightly change the presented bounds). In this case, if  $\ell$  is convex and bounded, then by (3) the regret of  $\mathcal{A}$  is bounded by  $\rho_{\mathcal{E}}(n) = \sqrt{n \ln N}$ . Setting  $\eta_t \equiv \phi \ln n / \sqrt{n}$  for some  $\phi > 0$  ( $\eta_t$  is independent of  $C(T)$  but depends on the time horizon  $n$ ), the bound (28) becomes, for  $g = O(1)$ ,

$$\begin{aligned} \widehat{L}_n - L_n(T, a) &\leq \sqrt{n(C(T) + 1) \left( \frac{\log n}{\lfloor \log(g+1) \rfloor} + 2 \right) \ln N} \\ &\quad + \frac{\phi \sqrt{n} \ln n}{8} + \frac{(C(T) + 1) \sqrt{n}}{\phi} \left( \frac{\log n}{\lfloor \log(g+1) \rfloor} + 2 \right) \\ &\quad + O\left(\frac{\sqrt{n}}{\ln n}\right). \end{aligned}$$

Furthermore, if an upper bound  $C$  on the complexity (number of switches) of the meta experts in the reference class is known in advance, then  $\eta_t$  can be set as a function of  $C \geq C(T)$  as well. Letting  $\eta_t \equiv \sqrt{8(C+1) \ln n \left( \frac{\log n}{\lfloor \log(g+1) \rfloor} + 2 \right)} / n$ , the bound (28) becomes

$$\begin{aligned} \widehat{L}_n - L_n(T, a) &\leq \sqrt{n(C(T) + 1) \left( \frac{\log n}{\lfloor \log(g+1) \rfloor} + 2 \right) \ln N} \\ &\quad + \sqrt{\frac{n(C+1) \left( \frac{\log n}{\lfloor \log(g+1) \rfloor} + 2 \right) \ln n}{2}} \\ &\quad + O\left(\sqrt{\frac{n}{(C+1) \ln n \left( \frac{\log n}{\lfloor \log(g+1) \rfloor} + 2 \right)}}\right). \end{aligned}$$

We note that these bounds are of order  $(C(T) + 1)\sqrt{n \ln^2 n}$ , respectively  $\sqrt{(C+1)n \ln^2 n}$ , only a factor of  $O(\sqrt{\ln n})$

larger than the ones of optimal order resulting from earlier algorithms [4], [5], [24] which have complexity  $O(n^2)$  (strictly speaking, the complexity of [4] is  $O(nN)$ , but, when combined with efficient algorithms designed for the base-expert class, only  $O(n^2)$  complexity versions are known [24]). In some applications, such as online quantization [24], the number of base experts  $N$  depends on the time horizon  $n$  in a polynomial fashion, that is,  $N \sim n^\beta$  for some  $\beta > 0$ . In such cases the order of the upper bound is not changed; it remains still  $O((C(T)+1)\sqrt{n \ln^2 n})$  if the number of switches is unknown, and  $O(\sqrt{(C(T)+1)n \ln^2 n})$  if the maximum number of switches  $C(T)$  is known in advance. This bound is within a factor of  $O(\sqrt{\ln n})$  of the best achievable regret for this case.

Next we observe that at the price of a slight increase of computational complexity, regret bounds of the optimal order can be obtained. Indeed, setting  $g = 2n^\gamma - 1$  for some  $\gamma \in (0, 1)$  and  $\eta_t \equiv \phi \sqrt{\frac{(2+1/\gamma) \ln n}{n}}$ ,  $\phi > 0$  independently of the maximum number of switches,

$$\begin{aligned} \hat{L}_n - L_n(T, a) &\leq \sqrt{n(C(T)+1) \ln N \left(\frac{1}{\gamma} + 2\right)} \\ &\quad + \left(\frac{\phi}{8} + \frac{C+1}{\phi}\right) \sqrt{\left(\frac{1}{\gamma} + 2\right) n \ln n} + O\left(\sqrt{\frac{n}{\ln n}}\right). \end{aligned}$$

If  $\eta_t$  is optimized for an a priori known bound  $C \geq C(T)$ , then we get

$$\begin{aligned} \hat{L}_n - L_n(T, a) &\leq \sqrt{n(C(T)+1) \left(\frac{1}{\gamma} + 2\right)} \left(\sqrt{\ln N} + \sqrt{\frac{\ln n}{2}}\right) \\ &\quad + O\left(\sqrt{\frac{n}{(C+1) \ln n}}\right). \end{aligned}$$

These bounds are of the same  $O((C(T)+1)\sqrt{n \ln n})$  and, respectively,  $O(\sqrt{(C+1)n \ln n})$  order as the ones achievable with the quadratic complexity algorithms [21], [24], but the complexity of our algorithm is only  $O(n^\gamma \ln n)$  times larger than that of running  $\mathcal{A}$  (which is typically linear in  $n$ ). Thus, in a sense the complexity of our algorithm can get very close to linear while guaranteeing a regret of optimal order. (Note however, that a factor  $1/\sqrt{\gamma}$  appears in the regret bounds so setting  $\gamma$  very small comes at a price.)

A similar behavior is observed for exp-concave loss functions. Indeed, if  $\ell$  is exp-concave and  $\mathcal{A}$  is the exponentially weighted average forecaster, then by (4) the regret of  $\mathcal{A}$  is bounded by  $\rho_{\mathcal{E}}(n) = \frac{\ln N}{\eta}$ . In this case, for  $g = O(1)$ , the bound (30) becomes

$$\begin{aligned} \hat{L}_n - L_n(T, a) &\leq \frac{(C(T)+1) \left(\frac{\log \frac{n}{C(T)+1}}{\log(g+1)} + 2\right)}{\eta} (\ln N + \ln n) + O(1). \end{aligned}$$

which is a factor of  $O(\ln n)$  larger than the existing optimal bounds of order  $O((C(T)+1) \ln n)$  (see [2]–[4], [6], [21]) valid for algorithms having complexity  $O(n^2)$  (again,

concerning [4], we mean its combination with some efficient algorithm designed for the base-expert class). Note that in this case the algorithm is strongly sequential as its parametrization is independent of the time horizon  $n$ . For  $g = 2n^\gamma - 1$ , we obtain a bound of optimal order  $O((C(T)+1) \ln n)$ :

$$\begin{aligned} \hat{L}_n - L_n(T, a) &\leq \frac{(C(T)+1) \left(\frac{1}{\gamma} + 2\right)}{\eta} (\ln N + \ln n) + O(1). \end{aligned}$$

Bounds of similar order can be obtained for exp-concave loss functions in the more general case when  $\mathcal{E}$  is not of size  $N$ , but is a bounded convex subset of an  $N$  dimensional linear space. Then  $\rho_{\mathcal{E}}(n) = O(\ln n)$  for several algorithms  $\mathcal{A}$  under different assumptions. This is the case for exp-concave loss functions when  $\mathcal{A}$  performs exponential weighting over all base experts. Using random-walk based sampling from log-concave distributions (see [32]), efficient probabilistic approximations exist to perform this weighting in many cases. Exact low complexity implementations, such as the Krichevsky-Trofimov estimate for the logarithmic loss [13] (see Example 1 below), are however, rare. When additional assumptions are made, e.g., the gradient of the loss function is bounded, the online Newton step algorithm of [12] can be applied to achieve logarithmic (standard) regret against the base-expert class  $\mathcal{E}$ . We refer to [33] for a survey.

#### F. The weight function $w^{KT}$

In this section we analyze the performance of Algorithm 1 for the case when the “Krichevsky-Trofimov” weight function  $w^{KT}$  is used. Our analysis is based on part (ii) of Lemma 3, following ideas of Willems and Krom [9] who only considered the logarithmic loss. Applying the weight function  $\hat{w}^{KT}$  (derived from  $w^{KT}$ ), this analysis improves the constants relative to Theorem 2 for small values of  $g$ , although the resulting bound has a less compact form. Nevertheless, in some special situations the bounds can be expressed in a simple form. This is the case for the logarithmic loss, where, for the special choice  $g = 1$ , applying (24), the new bound now achieves that of [9] proved for the same algorithm. The idea is that in the proof of Theorem 2 the concavity of  $\rho_{\mathcal{E}}$  was used to get simple bounds on sums which are sharp if the segments are of (approximately) equal length. However, in our construction the length of the sub-segments (corresponding to the same segment of the original transition path), or more precisely, their lower bounds, grow exponentially according to (25). This makes it possible to improve the upper bounds in Theorem 2. It is interesting to note that the weight functions  $w^{\mathcal{L}_1}$  and  $w^{\mathcal{L}_2}$  give better bounds for  $g = n^\gamma$ , where the segment lengths are approximately equal, while the large differences in the segment lengths for  $g = O(1)$  can be exploited by the weight function  $w^{KT}$ .

To obtain “almost closed-form” regret bounds for a general  $\rho_{\mathcal{E}}$ , we need the following technical lemma.

*Lemma 5:* Assume  $f : [1, \infty) \rightarrow (0, \infty)$  is a differentiable

function and  $G \geq 1$ . Define  $F : [1, \infty) \rightarrow [0, \infty)$  by

$$F(s) = \int_0^{\frac{\log s}{G}} f\left(\frac{s}{2^{cG}}\right) dc$$

for all  $s \geq 1$ . Then the second derivative of  $F$  is given by

$$F''(s) = \frac{f'(s)}{sG \ln 2} - \frac{f(s)}{s^2 G \ln 2}.$$

Therefore,  $F$  is concave on  $[1, \infty)$  if  $sf'(s) \leq f(s)$  for all  $s \geq 1$ .

*Proof:* First note that, since  $2^{cG} = s$  for  $c = \frac{\log s}{G}$ , Leibniz's integral rule gives

$$\begin{aligned} F'(s) &= \frac{f(1)}{sG \ln 2} + \int_0^{\frac{\log s}{G}} f'\left(\frac{s}{2^{cG}}\right) 2^{-cG} dc \\ &= \frac{f(1) - f(1) + f(s)}{sG \ln 2} = \frac{f(s)}{sG \ln 2} \end{aligned}$$

since

$$-\frac{\partial}{\partial c} \frac{f(2^{-cG})}{sG \ln 2} = f'(2^{-cG}) 2^{-cG}.$$

Differentiating  $F'$  gives the desired result.  $\blacksquare$

Next we give an improvement of Theorem 2 for small values of  $g$ . For simplicity, the bounds are only given for the tracking regret. It is much more complicated to obtain sharp bounds for the adaptive regret, since, similarly to the proof of Theorem 2, it would require to lower bound the probability that a new segment is started at some time instant  $t$ , but here the switch probabilities  $p_{KT}(t|t')$ , defined in (13), depend both on  $t$  and  $t'$ , unlike  $p_{L_1}(t|t')$  which only depends on  $t$ .

*Theorem 3:* Assume  $\rho_{\mathcal{E}}(x)$  is differentiable and satisfies  $\rho_{\mathcal{E}}(x) \geq x\rho'_{\mathcal{E}}(x)$  for all  $x \geq 1$ , and Algorithm 1 is run with weight function  $\{\hat{w}_t^{KT}\}$ . Let

$$\begin{aligned} S(C, n) &= (C+1) \int_0^{\frac{\log \frac{n}{C+1}}{\lfloor \log(g+1) \rfloor}} \rho_{\mathcal{E}}\left(\frac{n}{C+1} 2^{-c \lfloor \log(g+1) \rfloor}\right) dc \\ &\quad + 2(C+1) \rho_{\mathcal{E}}\left(\frac{n}{C+1}\right) \end{aligned}$$

and

$$\begin{aligned} \bar{r}_n(C) &= \frac{(C+1) \ln 2}{4} \left( \frac{\log^2 \frac{n}{C+1}}{\lfloor \log(g+1) \rfloor} \right. \\ &\quad \left. + \left( 4 + \frac{4}{\lfloor \log(g+1) \rfloor} \right) \log \frac{n}{C+1} + \lfloor \log(g+1) \rfloor + 8 \right). \end{aligned}$$

If  $\ell$  is convex in its first argument and takes values in the interval  $[0, 1]$ , and  $\eta_{t+1} \leq \eta_t$  for  $t = 1, \dots, n-1$ , then for any  $T \in \mathcal{T}_n$  the tracking regret satisfies, for all  $n$ ,

$$\hat{L}_n - L_n(T, a) \leq S(C, n) + \sum_{t=1}^n \frac{\eta_t}{8} + \frac{\bar{r}_n(C)}{\eta_n}. \quad (34)$$

On the other hand, if  $\ell$  is exp-concave for the value of  $\eta$  and  $\eta_t = \eta$  for  $t = 1, \dots, n$  in Algorithm 1, then for any  $T \in \mathcal{T}_n$  the tracking regret satisfies

$$\hat{L}_n - L_n(T, a) \leq S(C, n) + \frac{\bar{r}_n(C)}{\eta_n}. \quad (35)$$

*Proof:* We proceed similarly to the proof of Theorem 2 by first applying Lemma 1. However, the resulting two terms are now bounded using Lemma 3 (ii) instead of Jensen's inequality, which allows us to make use of the potentially large differences in the segment lengths.

For any transition path  $T = (t_1, \dots, t_C) \in \mathcal{T}_n$  let  $\hat{T} = (\hat{t}_1, \dots, \hat{t}_{\hat{C}}) \in \mathcal{T}_n$  denote the transition path defined by Lemma 3 with  $\hat{w}_n^{KT}(\hat{T}) > 0$ . The first term of the first upper bound given in Lemma 1 can be bounded as follows: for any segment  $[t_c, t_{c+1}] = [\hat{t}_{\hat{c}}, \hat{t}_{\hat{c}'}]$  of  $T$ , Lemma 3 (i) and (23) yield

$$\begin{aligned} &\sum_{i=\hat{c}}^{\hat{c}'-1} \rho_{\mathcal{E}}(\hat{t}_{i+1} - \hat{t}_i) \\ &\leq \int_0^{\frac{\log(t_{c+1} - t_c)}{\lfloor \log(g+1) \rfloor}} \rho_{\mathcal{E}}\left(\frac{t_{c+1} - t_c}{2^{c \lfloor \log(g+1) \rfloor}}\right) dc + 2\rho_{\mathcal{E}}(t_{c+1} - t_c). \end{aligned}$$

Since the right-hand side of the above inequality is a concave function of  $s = t_{c+1} - t_c$  by Lemma 5 and the conditions on  $\rho_{\mathcal{E}}$ , Jensen's inequality implies

$$\begin{aligned} &\sum_{i=0}^{\hat{C}} \rho_{\mathcal{E}}(\hat{t}_{i+1} - \hat{t}_i) \\ &= \sum_{c=0}^C \sum_{i=\hat{c}}^{\hat{c}'-1} \rho_{\mathcal{E}}(\hat{t}_{i+1} - \hat{t}_i) \\ &\leq \sum_{c=0}^C \left( \int_0^{\frac{\log(t_{c+1} - t_c)}{\lfloor \log(g+1) \rfloor}} \rho_{\mathcal{E}}\left(\frac{t_{c+1} - t_c}{2^{c \lfloor \log(g+1) \rfloor}}\right) dc + 2\rho_{\mathcal{E}}(t_{c+1} - t_c) \right) \\ &\leq (C+1) \int_0^{\frac{\log \frac{n}{C+1}}{\lfloor \log(g+1) \rfloor}} \rho_{\mathcal{E}}\left(\frac{n}{C+1} \cdot 2^{-c \lfloor \log(g+1) \rfloor}\right) dc \\ &\quad + 2(C+1) \rho_{\mathcal{E}}\left(\frac{n}{C+1}\right). \end{aligned} \quad (36)$$

The weight function can be bounded in a similar way. By the standard bound (14) on the Krichevsky-Trofimov estimate [14], we have

$$\begin{aligned} \ln \frac{1}{\hat{w}_n^{KT}(\hat{T})} &\leq \ln \frac{1}{w_n^{KT}(\hat{T})} \\ &\leq \sum_{c=0}^{\hat{C}} \left( \frac{1}{2} \ln(\hat{t}_{c+1} - \hat{t}_c) + \ln 2 \right). \end{aligned} \quad (37)$$

Applying (22) for a segment  $[t_c, t_{c+1}] = [\hat{t}_{\hat{c}}, \hat{t}_{\hat{c}'}]$  of  $T$  yields

$$\begin{aligned} &\sum_{i=\hat{c}}^{\hat{c}'-1} \left( \frac{1}{2} \ln(\hat{t}_{i+1} - \hat{t}_i) + \ln 2 \right) \\ &\leq \sum_{i=0}^{\left\lceil \frac{\log(t_{c+1} - t_c)}{\lfloor \log(g+1) \rfloor} \right\rceil - 1} \left( \frac{1}{2} \ln \left( \frac{t_{c+1} - t_c}{2^{i \lfloor \log(g+1) \rfloor}} \right) + \ln 2 \right) \\ &\quad + \frac{1}{2} \ln(t_{c+1} - t_c) + \ln 2 \end{aligned}$$

$$\begin{aligned}
&= \frac{\ln 2}{2} \left\lceil \frac{\log(t_{c+1} - t_c)}{\lfloor \log(g+1) \rfloor} \right\rceil \times \\
&\quad \times \left( \log(t_{c+1} - t_c) - \frac{\left\lceil \frac{\log(t_{c+1} - t_c)}{\lfloor \log(g+1) \rfloor} \right\rceil - 1}{2} \lfloor \log(g+1) \rfloor + 2 \right) \\
&\quad + \frac{1}{2} \ln(t_{c+1} - t_c) + \ln 2 \\
&\leq \frac{\ln 2}{4} \left( \frac{\log^2(t_{c+1} - t_c)}{\lfloor \log(g+1) \rfloor} + \lfloor \log(g+1) \rfloor + 8 \right. \\
&\quad \left. + \left( 4 + \frac{4}{\lfloor \log(g+1) \rfloor} \right) \log(t_{c+1} - t_c) \right) \\
&\leq 2\sqrt{(C(T)+1)n \ln N} \left( 1 + \frac{1 - \sqrt{\frac{C+1}{n}}}{\gamma \ln n} \right) \\
&\quad + \frac{\phi + \frac{C+1}{\phi}}{8} \sqrt{2n \ln n \left( 4 + \gamma + \frac{1}{\gamma} \right)} + O\left(\sqrt{\frac{n}{\ln n}}\right)
\end{aligned}$$

where in the last step we bounded the ceiling function from above and from below, as appropriate. Furthermore, it is easy to check that the last expression above is concave in  $s = t_{c+1} - t_c$ . Therefore, combining it with (37), applying Jensen's inequality, we obtain

$$\ln \frac{1}{\hat{w}_n^{KT}(\hat{T})} \leq \bar{r}_n(C).$$

Applying this bound and (36) in Lemma 1 yields the statements of the theorem.  $\blacksquare$

We now apply Theorem 3 to the exponentially weighted average predictor. For bounded convex loss functions we have  $\rho_{\mathcal{E}}(n) = \sqrt{n \ln N}$ . Assuming  $g = O(1)$ , if  $\eta_t \equiv \phi \sqrt{\frac{2 \ln 2}{n \lfloor \log(g+1) \rfloor}} \log n$ ,  $\phi > 0$  (i.e.,  $\eta_t$  is independent of the number of switches  $C(T)$ ), we obtain

$$\begin{aligned}
&\hat{L}_n - L_n(T, a) \\
&\leq 2\sqrt{(C(T)+1)n \ln N} \left( 1 + \frac{1 - \sqrt{\frac{C+1}{n}}}{\lfloor \log(g+1) \rfloor \ln 2} \right) \\
&\quad + \frac{\phi + \frac{C+1}{\phi}}{4} \log n \sqrt{\frac{n \ln 2}{2 \lfloor \log(g+1) \rfloor}} + o((C+1)\sqrt{n}).
\end{aligned}$$

Optimizing  $\eta_t$  as a function of an upper bound  $C$  on the number of switches yields

$$\begin{aligned}
&\hat{L}_n - L_n(T, a) \\
&\leq 2\sqrt{(C(T)+1)n \ln N} \left( 1 + \frac{1 - \sqrt{\frac{C+1}{n}}}{\lfloor \log(g+1) \rfloor \ln 2} \right) \\
&\quad + \sqrt{\frac{(C+1)n \log^2 \frac{n}{C+1} \ln 2}{8 \lfloor \log(g+1) \rfloor}} + o(\sqrt{(C+1)n}).
\end{aligned}$$

Note that if  $N = O(n^\beta)$  for some  $\beta > 0$ , the first term is asymptotically negligible compared to the second in the above bounds. For example, if  $\eta$  is set independently of  $C$ , we obtain

$$\begin{aligned}
&\hat{L}_n - L_n(T, a) \\
&\leq \frac{\phi + \frac{C+1}{\phi}}{4} \log n \sqrt{\frac{n \ln 2}{2 \lfloor \log(g+1) \rfloor}} + o((C+1)\sqrt{n}).
\end{aligned}$$

On the other hand, if  $g = 2n^\gamma - 1$ , the bound becomes

$$\hat{L}_n - L_n(T, a)$$

when  $\eta$  is set independently of  $C$ .

For exp-concave loss functions we have, for  $g = O(1)$ ,

$$\begin{aligned}
&\hat{L}_n - L_n(T, a) \\
&\leq \frac{C+1}{4\eta} \left( \frac{\log \frac{n}{C+1}}{\lfloor \log(g+1) \rfloor} + 2 \right) \left( 4 \ln N + \ln \frac{n}{C+1} \right) \\
&\quad + O(C \ln n)
\end{aligned}$$

while if  $g = 2n^\gamma - 1$  we get

$$\begin{aligned}
&\hat{L}_n - L_n(T, a) \\
&\leq \frac{C+1}{4\eta} \left( 4 \left( \frac{1}{\gamma} + 2 \right) \ln N + \left( 4 + \gamma + \frac{1}{\gamma} \right) \ln n \right) \\
&\quad + O(C).
\end{aligned}$$

Note that for both types of loss functions we have a clear improvement relative to Theorem 2, where we used the weight function  $w^{\mathcal{L}_1}$ , for the case when  $g = O(1)$ . However, no such distinction can be made for  $g = 2n^\gamma - 1$ . Indeed, for convex loss functions constant multiplicative changes in  $\eta$  vary the exact form of the factor  $(C+a)/b$ , with constants  $a, b > 0$  in the second term, and, consequently, the order of the bounds depends on the relative size of  $C$ , while, for example, the value of  $\eta$  determines the order of the bounds for exp-concave losses, e.g., constructing the weigh function  $\hat{w}$  from  $w^{\mathcal{L}_1}$  is better for  $\gamma \geq 1/3$ . Also note that the above bounds for  $g = 3$  and  $g = 4$  have improved leading constant compared to [10] and [31], respectively.

#### IV. RANDOMIZED PREDICTION

The results of the previous section may be adapted to the closely related model of randomized prediction. In this framework, the decision maker plays a repeated game against an adversary as follows: at each time instant  $t = 1, \dots, n$ , the decision maker chooses an action  $I_t$  from a finite set, say  $\{1, \dots, N\}$  and, independently, the adversary assigns losses  $\ell_{i,t} \in [0, 1]$  to each action  $i = 1, \dots, n$ . The goal of the decision maker is to minimize the cumulative loss  $\hat{L}_n = \sum_{t=1}^n \ell_{I_t, t}$ .

Similarly to the previous section, the decision maker may try to compete with the best sequence of actions that can change actions a limited number of times. More precisely, the set of base experts is  $\mathcal{E} = \{1, \dots, N\}$  and as before, we may define a meta expert that changes base experts  $C$  times by a transition path  $T = (t_1, \dots, t_C; n)$  and a vector of actions  $a = (i_0, \dots, i_C)$ , where  $t_0 := 1 < t_1 < \dots < t_C < t_{C+1} := n+1$  and  $i_j \in \{1, \dots, N\}$ . The total loss of the meta expert indexed by  $(T, a)$ , accumulated during  $n$  rounds, is

$$L_n(T, a) = \sum_{c=0}^C L_{i_c}(t_c, t_{c+1})$$

with

$$L_{i_c}(t_c, t_{c+1}) = \sum_{t=t_c}^{t_{c+1}-1} \ell_{i_c, t}.$$

There are two differences relative to the setup considered earlier. First, we do not assume that the loss function satisfies special properties such as convexity in the first argument (although we do require that it be bounded). Second, we do not assume in the current setup that the action space is convex, and so a convex combination of the experts' advice is not possible. On the other hand, similar results as before can be achieved if the decision maker may randomize its decisions, and in this section we deal with this situation.

In randomized prediction, before taking an action, the decision maker chooses a probability distribution  $\mathbf{p}_t$  over  $\{1, \dots, N\}$  (a vector in the probability simplex  $\Delta_N$  in  $\mathbb{R}^N$ ), and chooses an action  $I_t$  distributed according to  $\mathbf{p}_t$  (conditionally, given the past actions of the decision maker and the losses assigned by the adversary).

Note that now both  $\hat{L}_n$  and  $L_n(T, a)$  are random variables not only because the decision takes randomized decisions but also because the losses set by the adversary may depend on past randomized choices of the decision maker. (This model is known as the “non-oblivious adversary”.) We may define the *expected loss* of the decision maker by

$$\bar{\ell}_t(\mathbf{p}_t) = \sum_{i=1}^N p_{i,t} \ell_{i,t}$$

where  $p_{i,t}$  denotes the  $i$ -th component of  $\mathbf{p}_t$ .

For details and discussion of this standard model we refer to [1, Section 4.1]. In particular, since the results presented in Section I can be extended to time-varying loss functions and since  $\bar{\ell}_t$  is a linear (convex) function, it can be shown that regret bounds of any forecaster in the model of Section I can be extended to the sequence of loss functions  $\bar{\ell}_t$ . That is, the bounds can be converted into bounds for the expected regret of a randomized forecaster. Furthermore, it is shown in [1, Lemma 4.1] how such bounds in expectation can be converted to bounds that hold with high probability.

For example, a straightforward combination of [1, Lemma 4.1] and Theorem 2 implies the following. Consider a prediction algorithm  $\mathcal{A}$  defined in the model of Section III-A, that chooses an action in the decision space  $\mathcal{D} = \Delta_N$  and suppose that it satisfies a regret bound of the form (5) under the loss function  $\bar{\ell}_t(\mathbf{p}_t)$ . Algorithm 2 below, which is a variant of Algorithm 1, converts  $\mathcal{A}$  into a forecaster under the randomized model. At each time instant  $t$ , the algorithm chooses, in a randomized way, a transition path  $T = (t_1, \dots, t_C; t) \in \mathcal{T}_t$ , and uses the distribution  $\mathbf{p}_{\mathcal{A},t}(\tau_t(T))$  that  $\mathcal{A}$  would choose, had it been started at time  $\tau_t(T)$ , the time of the last change in the path  $T$  up to time  $t$ . In the definition of the algorithm

$$\bar{L}_t(\mathcal{A}, T) = \sum_{c=0}^C \bar{L}_{\mathcal{A}}(t_c, t_{c+1})$$

denotes the cumulative expected loss of algorithm  $\mathcal{A}$ , where

we define  $t_0 = 1$  and  $t_{c+1} = t + 1$ , and

$$\bar{L}_{\mathcal{A}}(t_c, t_{c+1}) = \sum_{s=t_c}^{t_{c+1}-1} \bar{\ell}_s(\mathbf{p}_{\mathcal{A},s}(t_c))$$

is the cumulative expected loss suffered by  $\mathcal{A}$  in the time interval  $[t_c, t_{c+1})$  with respect to  $\bar{\ell}_s$  for  $s \in [t_c, t_{c+1})$ .

---

**Algorithm 2** Randomized tracking algorithm.

---

**Input:** Prediction algorithm  $\mathcal{A}$ , weight function  $\{w_t; t = 1, \dots, n\}$ , learning parameters  $\eta_t > 0, t = 1, \dots, n$ . For  $t = 1, \dots, n$  choose  $T \in \mathcal{T}_t$  according to the distribution

$$q_t(T) = \frac{w_t(T) e^{-\eta_t \bar{L}_{t-1}(\mathcal{A}, T_{t-1})}}{\sum_{T' \in \mathcal{T}_t} w_t(T') e^{-\eta_t \bar{L}_{t-1}(\mathcal{A}, T'_{t-1})}},$$

choose  $\mathbf{p}_t = \mathbf{p}_{\mathcal{A},t}(\tau_t(T))$ , and pick  $I_t \sim \mathbf{p}_t$ .

---

*Corollary 1:* Suppose  $\ell_{i,t} \in [0, 1]$  for all  $i = 1, \dots, N$  and  $t = 1, \dots, n$ , and  $\mathcal{A}$  satisfies (5) with respect to the loss function  $\{\ell_t\}$ . Assume Algorithm 2 is run with weight function  $\{\hat{w}^{\mathcal{L}_1}\}$  for some  $\epsilon > 0$ . Let  $\delta \in (0, 1)$ . For any  $T \in \mathcal{T}_n$ , the regret of the algorithm satisfies, with probability at least  $1 - \delta$ ,

$$\begin{aligned} \hat{L}_n - L_n(T, a) &\leq L_{C(T),n}(C(T) + 1) \rho \epsilon \left( \frac{n}{L_{C(T),n}(C(T) + 1)} \right) + \sum_{t=1}^n \frac{\eta_t}{8} \\ &\quad + \frac{r_n(L_{C(T),n}(C(T) + 1) - 1)}{\eta_n} + \sqrt{\frac{n}{2} \ln \frac{1}{\delta}}. \end{aligned}$$

where  $r_n(C)$  and  $L_{C,n}$  are defined as in Theorem 2.

*Proof:* First note that Theorem 2 can easily be extended to time-varying loss functions (in fact, Lemma 1, and consequently Theorem 2, uses the bound (2) which allows time-varying loss functions). Combining the obtained bound for the expected loss with [1, Lemma 4.1] proves the corollary. ■

## V. EXAMPLES

In this section we apply the results of the paper for a few specific examples.

*Example 1 (Krichevsky-Trofimov mixtures):* Assume  $\mathcal{D} = \mathcal{E} = (0, 1)$  and  $\mathcal{Y} = \{0, 1\}$ , and consider the logarithmic loss defined as  $\ell(p, y) = -\mathbb{I}_{y=1} \ln p - \mathbb{I}_{y=0} \ln(1-p)$ . As mentioned before, the logarithmic loss is exp-concave with  $\eta \leq 1$ , and hence we choose  $\eta = 1$ . This loss plays a central role in data compression. In particular, if a prediction method achieves, on a particular binary sequence  $y^n = (y_1, \dots, y_n)$ , a loss  $\hat{L}_n$ , then using arithmetic coding the sequence can be described with at most  $\hat{L}_n + 2$  bits [34]. We note that the choice of the expert class  $\mathcal{E} = (0, 1)$  corresponds to the situation where the sequence  $y^n$  is encoded using an i.i.d. coding distribution. Competing against the expert class  $\mathcal{E} = (0, 1)$  also has a probabilistic interpretation: it is equivalent to minimizing the worst case maximum coding redundancy relative to the class of i.i.d. source distributions on  $\{0, 1\}^n$ .

Let  $n_0(t) = \sum_{s=1}^t \mathbb{I}_{y_s=0}$  and  $n_1(t) = \sum_{s=1}^t \mathbb{I}_{y_s=1}$  denote the number of 0s and 1s in  $y^t$ , respectively. Then the loss of an expert  $\theta \in (0, 1)$  at time  $t$  is

$$\begin{aligned} L_{\theta,t} &= -\ln\left((1-\theta)^{n_0(t)}\theta^{n_1(t)}\right) \\ &= -n_0(t)\ln(1-\theta) - n_1(t)\ln\theta \end{aligned}$$

which is the negative log-probability assigned to  $y^t$  by a memoryless binary Bernoulli source generating 1s with probability  $\theta$ . The Krichevsky-Trofimov forecaster is an exponentially weighted average forecaster over all experts  $\theta \in \mathcal{E}$  using initial weights  $1/(\pi\sqrt{\theta(1-\theta)})$  (i.e., the Beta(1/2, 1/2) distribution) defined as

$$\begin{aligned} p_t^{KT}(y^{t-1}) &= \int_0^1 \frac{e^{-L_{\theta,t-1}}}{\pi\sqrt{\theta(1-\theta)}} d\theta \\ &= \int_0^1 \frac{(1-\theta)^{n_0(t-1)}\theta^{n_1(t-1)}}{\pi\sqrt{\theta(1-\theta)}} d\theta. \end{aligned}$$

It is well known that  $p_t^{KT}$  can be computed efficiently as  $p_t^{KT}(1|y^{t-1}) = 1 - p_t^{KT}(0|y^{t-1}) = \frac{n_1(t-1)+1/2}{t}$ . By [14], the performance of the Krichevsky-Trofimov mixture forecaster can be bounded as

$$R_n \leq \frac{1}{2} \ln n + \ln 2.$$

In this framework, a meta expert based on the base expert class  $\mathcal{E}$  is allowed to change  $\theta \in \mathcal{E}$  a certain number of times. In the probabilistic interpretation, this corresponds to the problem of coding a piecewise i.i.d. source [2], [3], [7]–[9]. If we apply Algorithm 1 to this problem with  $\hat{w}^{KT}$ , we can improve upon Theorem 3 by using  $\bar{r}_n(C)$  instead of  $S(C, n)$  in the bound (note that  $\bar{r}_n(C)$  was obtained by calculating the Krichevsky-Trofimov bound for the transition probabilities), and obtain, for any transition path  $T \in \mathcal{T}_n$  and meta expert  $(T, a)$

$$\begin{aligned} \hat{L}_n - L_n(T, a) &\leq 2\bar{r}_n(C(T)) \\ &= \frac{(C(T) + 1) \ln 2}{2} \frac{\log^2 \frac{n}{C(T)+1}}{[\log(g+1)]} + O((C(T) + 1) \ln n). \end{aligned}$$

For  $g = 1$ , this bound recovers that of [9] (at least in the leading term), and improves the leading constant for  $g = 3$  and  $g = 4$  when compared to [10] and [11], respectively.

On the other hand, for  $g = 2n^\gamma - 1$ ,  $\gamma > 0$ , using with  $\hat{w}^{\mathcal{L}_1}$  in Algorithm 1, Theorem 3 implies

$$\hat{L}_n - L_n(T, a) \leq \frac{3(C(T) + 1)}{2} \left( \frac{1}{\gamma} + 2 \right) \ln n + O(1).$$

This bound achieves the optimal  $O(\ln n)$  order for any  $\gamma > 0$ ; however, with increased leading constant. On the negative side, for specific choices of  $\gamma$  our algorithm does not recover the best leading constants known in the literature (partly due to the common bounding technique for all  $\gamma$ ): If  $\gamma = 1/2$ , our bound is a constant factor worse than those of [7] and [8] which have the same  $O(n^{3/2})$  complexity (disregarding logarithmic factors); on the other hand, in case  $\gamma = 1$  our algorithm is identical to the  $O(n^2)$  complexity algorithm of Shamir and Merhav [3], and hence an optimal bound can be proved for

$\hat{w}^{\mathcal{L}_1}$  (and for  $\hat{w}^{\mathcal{L}_2}$ ), as done in [3] achieving Merhav's lower bound [30].

*Example 2 (Tracking structured classes of base experts):*

In recent years a significant body of research has been devoted to prediction problems in which the forecaster competes with a large but structured class of experts. We refer to [1], [16], [17], [24], [35]–[38] for an incomplete but representative list of papers. A quite general framework that has been investigated is the following: a base expert is represented by a  $d$ -dimensional binary vector  $v \in \{0, 1\}^d$ . Let  $\mathcal{E} \subset \{0, 1\}^d$  be the class of experts. The decision space  $\mathcal{D}$  is the convex hull of  $\mathcal{E}$ , so the forecaster chooses, at each time instant  $t = 1, \dots, n$ , a convex combination  $\hat{p}_t = \sum_{v \in \mathcal{E}} \pi_{v,t} v \in \mathcal{D} \subset [0, 1]^d$ . The outcome space is  $\mathcal{Y} = [0, 1]^d$  and if the outcome is  $y_t \in \mathcal{Y}$ , then the loss of expert  $v$  is  $\ell(v, y_t) = v^T y_t$ , the standard inner product of  $v$  and  $y_t$ . The loss of the forecaster equals  $\ell(\hat{p}_t, y_t) = \sum_{v \in \mathcal{E}} \pi_{v,t} v^T y_t$ . [36] introduces a general prediction algorithm, called “Component Hedge,” that achieves a regret

$$\begin{aligned} \sum_{t=1}^n \ell(\hat{p}_t, y_t) - \min_{v \in \mathcal{E}} \sum_{t=1}^n \ell(v, y_t) \\ \leq d\sqrt{2Kn \ln(d/K)} + dK \ln(d/K) \end{aligned}$$

where  $K = \max_{v \in \mathcal{E}} \|v\|_1$ . What makes Component Hedge interesting, apart from its good regret guarantee, is that for many interesting classes of base experts it can be calculated in time that is polynomial in  $d$ , even when  $\mathcal{E}$  has exponentially many experts. We refer to [36] for a list of such examples. The results of this paper show that we may obtain efficiently computable algorithms for tracking such structured classes of base experts. For example, (28) of Theorem 2 applies in this case, with  $\rho_{\mathcal{E}}(n) = d\sqrt{2Kn \ln(d/K)} + dK \ln(d/K)$ . The calculations of Section III-E may be easily modified for this case in a straightforward manner.

*Example 3 (Tracking the best quantizers):* The problem of limited-delay adaptive universal lossy source coding of individual sequences has recently been investigated in detail [18]–[20], [24], [39]–[41]. In the widely used model of fixed-rate lossy source coding at rate  $R$ , an infinite sequence of  $[0, 1]$ -valued source symbols  $x_1, x_2, \dots$  is transformed into a sequence of channel symbols  $y_1, y_2, \dots$  which take values from the finite channel alphabet  $\{1, 2, \dots, M\}$ ,  $M = 2^R$ , and these channel symbols are then used to produce the ( $[0, 1]$ -valued) reproduction sequence  $\hat{x}_1, \hat{x}_2, \dots$ . The quality of the reproduction is measured by the average distortion  $\sum_{t=1}^n d(x_t, \hat{x}_t)$ , where  $d$  is some nonnegative bounded distortion measure. The squared error  $d(x, x') = (x - x')^2$  is perhaps the most popular example.

The scheme is said to have overall delay at most  $\delta$  if there exist nonnegative integers  $\delta_1$  and  $\delta_2$  with  $\delta_1 + \delta_2 \leq \delta$  such that each channel symbol  $y_n$  depends only on the source symbols  $x_1, \dots, x_{n+\delta_1}$  and the reproduction  $\hat{x}_n$  for the source symbol  $x_n$  depends only on the channel symbols  $y_1, \dots, y_{n+\delta_2}$ . When  $\delta = 0$ , the scheme is said to have zero delay. In this case,  $y_n$  depends only on  $x_1, \dots, x_n$ , and  $\hat{x}_n$  on  $y_1, \dots, y_n$ , so that the encoder produces  $y_n$  as soon as  $x_n$  becomes available, and

the decoder can produce  $\hat{x}_n$  when  $y_n$  is received. The natural reference class of codes (experts) in this case is the set of  $M$ -level scalar quantizers

$$\mathcal{Q} = \{Q : [0, 1] \rightarrow \{c_1, \dots, c_M\}, \{c_1, \dots, c_M\} \subset [0, 1]\}.$$

The relative loss with respect to the reference class  $\mathcal{Q}$  is known in this context as the distortion redundancy. For the squared error distortion, the best randomized coding methods [20], [39], [41], with linear computational complexity with respect to the set  $\mathcal{Q}$ , yield a distortion redundancy of order  $O(n^{-1/4}\sqrt{\ln n})$ .

The problem of competing with the best time-variant quantizer that can change the employed quantizer several times (i.e., tracking the best quantizer), was analyzed in [24], based on a combination of [20] and the tracking algorithm of [4]. There the best linear-complexity scheme achieves  $O((C+1)\ln n/n^{1/6})$  distortion redundancy when an upper bound  $C$  on the number of switches in the reference class is known in advance. On the other hand, applying our scheme with  $g = O(1)$  in the method of [24] and using the bounds in Section III-E, gives a linear-complexity algorithm with distortion redundancy  $O((C+1)^{1/2}\ln^{3/4}(n)/n^{1/4}) + O((C+1)/(\ln^{1/2}(n)/n^{1/2}))$  if  $C$  is known in advance and only slightly worse  $O((C+1)^{1/2}\ln^{3/4}(n)/n^{1/4}) + O((C+1)\ln(n)/n^{1/2})$  distortion redundancy if  $C$  is unknown. When  $g = 2n^\gamma - 1$ , the distortion redundancy for linear complexity becomes somewhat worse, proportional to  $n^{-\frac{1}{2(2+\gamma)}}$  up to logarithmic factors.

## VI. CONCLUSION

We examined the problem of efficiently tracking large expert classes where the goal of the predictor is to perform as well as a given reference class. We considered prediction strategies that compete with the class of switching strategies that can segment a given sequence into several blocks, and follow the advice of a different base expert in each block. We derived a family of efficient tracking algorithms that, for any prediction algorithm  $\mathcal{A}$  designed for the base class, can be implemented with time and space complexity  $O(n^\gamma \ln n)$  times larger than that of  $\mathcal{A}$ , where  $n$  is the time horizon and  $\gamma \geq 0$  is a parameter of the algorithm. With  $\mathcal{A}$  properly chosen, our algorithm achieves a regret bound of optimal order for  $\gamma > 0$ , and only  $O(\ln n)$  times larger than the optimal order for  $\gamma = 0$  for all typical regret bound types we examined. For example, for predicting binary sequences with switching parameters, our method achieves the optimal  $O(\ln n)$  regret rate with time complexity  $O(n^{1+\gamma} \ln n)$  for any  $\gamma \in (0, 1)$ . Linear complexity algorithms that achieve optimal regret rate for small base expert classes have been shown to exist in [4] and [6]. Our results show that the optimal rate is achievable with the slightly larger  $O(n^{1+\gamma} \ln n)$ ,  $\gamma > 0$ , complexity even if the number of switches is not known in advance and the base expert class is large. It remains, however, an open question whether the optimal rate is achievable with a linear complexity algorithm in this case.

## ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their comments that helped improve the presentation of the paper.

## REFERENCES

- [1] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge: Cambridge University Press, 2006.
- [2] F. M. J. Willems, "Coding for a binary independent piecewise-identically-distributed source," *IEEE Transactions on Information Theory*, vol. IT-42, pp. 2210–2217, Nov. 1996.
- [3] G. I. Shamir and N. Merhav, "Low-complexity sequential lossless coding for piecewise-stationary memoryless sources," *IEEE Transactions on Information Theory*, vol. IT-45, pp. 1498–1519, July 1999.
- [4] M. Herbster and M. K. Warmuth, "Tracking the best expert," *Machine Learning*, vol. 32, no. 2, pp. 151–178, 1998.
- [5] V. Vovk, "Derandomizing stochastic prediction strategies," *Machine Learning*, vol. 35, no. 3, pp. 247–282, Jun. 1999.
- [6] W. Koolen and S. de Rooij, "Combining expert advice efficiently," in *Proceedings of the 21st Annual Conference on Learning Theory, COLT 2008*, Helsinki, Finland, July 2008, pp. 275–286.
- [7] C. Monteleoni and T. S. Jaakkola, "Online learning of non-stationary sequences," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004.
- [8] S. de Rooij and T. van Erven, "Learning the switching rate by discretising Bernoulli sources online," in *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS 2009)*, ser. JMLR Workshop and Conference Proceedings, vol. 5, Clearwater Beach, Florida USA, April 2009, pp. 432–439.
- [9] F. Willems and M. Krom, "Live-and-die coding for binary piecewise i.i.d. sources," in *Proceedings of the 1997 IEEE International Symposium on Information Theory (ISIT 1997)*, Ulm, Germany, June–July 1997, p. 68.
- [10] A. György, T. Linder, and G. Lugosi, "Efficient tracking of the best of many experts," in *Information and Communication Conference*, Budapest, Aug. 25–28 2008, pp. 3–4.
- [11] E. Hazan and C. Seshadhri, "Efficient learning algorithms for changing environments," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 393–400.
- [12] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning Journal*, vol. 69, no. 2–3, pp. 169–192, 2007.
- [13] R. E. Krichevsky and V. K. Trofimov, "The performance of universal encoding," *IEEE Transactions on Information Theory*, vol. IT-27, pp. 199–207, Mar. 1981.
- [14] Y. M. Shtarkov, "Universal sequential coding of single messages," *Problems of Information Transmission*, vol. 23, pp. 3–17, 1987.
- [15] F. M. J. Willems, Y. N. Shtarkov, and T. J. Tjalkens, "The context-tree weighting method: Basic properties," *IEEE Transactions on Information Theory*, vol. IT-41, pp. 653–664, May 1995.
- [16] A. Kalai and S. Vempala, "Efficient algorithms for the online decision problem," in *Proceedings of the 16th Annual Conference on Learning Theory and the 7th Kernel Workshop, COLT-Kernel 2003*, B. Schölkopf and M. Warmuth, Eds. New York, USA: Springer, Aug. 2003, pp. 26–40.
- [17] E. Takimoto and M. K. Warmuth, "Path kernels and multiplicative updates," *Journal of Machine Learning Research*, vol. 4, pp. 773–818, 2003.
- [18] T. Linder and G. Lugosi, "A zero-delay sequential scheme for lossy coding of individual sequences," *IEEE Transactions on Information Theory*, vol. 47, pp. 2533–2538, Sep. 2001.
- [19] T. Weissman and N. Merhav, "On limited-delay lossy coding and filtering of individual sequences," *IEEE Transactions on Information Theory*, vol. 48, pp. 721–733, Mar. 2002.
- [20] A. György, T. Linder, and G. Lugosi, "Efficient algorithms and minimax bounds for zero-delay lossy source coding," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2337–2347, Aug. 2004.
- [21] S. Kozat and A. Singer, "Universal switching linear least squares prediction," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 189–204, Jan. 2008.
- [22] —, "Switching strategies for sequential decision problems with multiplicative loss with application to portfolios," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2192–2208, June 2009.



- [23] —, “Universal randomized switching,” *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1922–1927, March 2010.
- [24] A. György, T. Linder, and G. Lugosi, “Tracking the best quantizer,” *IEEE Transactions on Information Theory*, vol. 54, pp. 1604–1625, Apr. 2008.
- [25] M. Zinkevich, “Online convex programming and generalized infinitesimal gradient ascent,” in *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, Washington, DC, USA, 2003.
- [26] M. Herbster and M. K. Warmuth, “Tracking the best linear predictor,” *Journal of Machine Learning Research*, vol. 1, pp. 281–309, 2001.
- [27] G. Stoltz and G. Lugosi, “Internal regret in on-line portfolio selection,” *Machine Learning*, vol. 59, pp. 125–159, 2005.
- [28] A. Blum and Y. Mansour, “From external to internal regret,” *Journal of Machine Learning Research*, vol. 8, pp. 1307–1324, Dec. 2007.
- [29] A. V. Chernov and F. Zhdanov, “Prediction with expert advice under discounted loss,” in *ALT*, 2010, pp. 255–269.
- [30] N. Merhav, “On the minimum description length principle for sources with piecewise constant parameters,” *IEEE Transactions on Information Theory*, pp. 1962–1967, November 1993.
- [31] E. Hazan and C. Seshadhri, “Adaptive algorithms for online decision problems,” *Electronic Colloquium on Computational Complexity*, Tech. Rep. 07-088, 2007.
- [32] L. Lovász and S. Vempala, “Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization,” in *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2006, pp. 57–68.
- [33] S. Bubeck, “Introduction to online optimization,” Lecture Notes, Princeton University, 2011, <http://www.princeton.edu/~sbubeck/BubeckLectureNotes.pdf>.
- [34] T. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 2006.
- [35] D. P. Helmbold and M. K. Warmuth, “Learning permutations with exponential weights,” *JMLR*, vol. 10, pp. 1705–1736, 2009.
- [36] W. M. Koolen, M. K. Warmuth, and J. Kivinen, “Hedging structured concepts,” in *23rd Annual Conference on Learning Theory*, 2010.
- [37] N. Cesa-Bianchi and G. Lugosi, “Combinatorial bandits,” *Journal of Computer and System Sciences*, vol. 78, pp. 1404–1422, 2012.
- [38] V. Dani, T. Hayes, and S. Kakade, “The price of bandit information for online optimization,” in *Proceedings of NIPS 2008.*, 2008.
- [39] A. György, T. Linder, and G. Lugosi, “A “follow the perturbed leader”-type algorithm for zero-delay quantization of individual sequences,” in *Proc. Data Compression Conference*, Snowbird, UT, USA, Mar. 2004, pp. 342–351.
- [40] S. Matloub and T. Weissman, “Universal zero delay joint source-channel coding,” *IEEE Transactions on Information Theory*, vol. 52, pp. 5240–5250, 2006.
- [41] A. György and G. Neu, “Near-optimal rates for limited-delay universal lossy source coding,” in *Proceedings of the IEEE International Symposium on Information Theory*, St. Petersburg, Russia, July-August 2011, pp. 2344–2348.