Tracking the best of many experts *

András György[1], Tamás Linder[2], and Gábor Lugosi[3]

[1] Informatics Laboratory, Computer and Automation Research Institute
of the Hungarian Academy of Sciences,
Lágymányosi u. 11, Budapest, Hungary, H-1111
`gya@szit.bme.hu`
[2] Department of Mathematics and Statistics,
Queen's University, Kingston, Ontario,
Canada K7L 3N6
`linder@mast.queensu.ca`
[3] Department of Economics, Universitat Pompeu Fabra
Ramon Trias Fargas 25-27, 08005 Barcelona, Spain
`lugosi@upf.es`

Abstract. An algorithm is presented for online prediction that allows to track the best expert efficiently even if the number of experts is exponentially large, provided that the set of experts has a certain structure allowing efficient implementations of the exponentially weighted average predictor. As an example we work out the case where each expert is represented by a path in a directed graph and the loss of each expert is the sum of the weights over the edges in the path.

1 Introduction

The basic theoretical results of prediction using expert advice were pioneered by Hannan [7] and Blackwell [2] in the 1950's and brought to the center of attention in learning theory in the 1990's by Vovk [16], Littlestone and Warmuth [11], Cesa-Bianchi, Freund, Helmbold, Haussler, Schapire, and Warmuth [4]. These results show that it is possible to construct algorithms for online prediction that predict an arbitrary sequence of outcomes almost as well as the best of N experts in the sense that the cumulative loss of the predictor is at most as large as that of the best expert plus a term proportional to $\sqrt{T \ln N}$ for any bounded loss function, where T is the number of rounds in the prediction game. The logarithmic dependence on the number of experts makes it possible to obtain meaningful bounds even if the pool of experts is very large. However, the basic prediction algorithms, such as the exponentially weighted average predictor, have a computational complexity proportional to the number of experts and are therefore infeasible when the number of experts is very large.

* This research was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada, the NATO Science Fellowship of Canada, the János Bolyai Research Scholarship of the Hungarian Academy of Sciences, Spanish Ministry of Science and Technology and FEDER, grant BMF2003-03324, and by the PASCAL Network of Excellence under EC grant no. 506778.

However, in many applications the set of experts has a certain structure that may be exploited to construct efficient prediction algorithms. Perhaps the best known such example is the problem of *tracking the best expert*. In this problem there is a small number of "base" experts and the goal of the predictor is to predict as well as the best of "meta" experts defined by any sequence of $m+1$ base experts and any partition of the time indexes up to T into $m+1$ contiguous blocks such that in block i a meta expert predicts according to the ith base expert in its defining sequence for $i = 0, \ldots, m$. If there are N base experts and the length of the prediction game is T then the total number of meta experts is $\sum_{k=0}^{m} \binom{T-1}{k} N(N-1)^k$. This problem was solved by Herbster and Warmuth [8] who exhibited computationally efficient algorithms that predict almost as well as the best of the meta experts and have regret bounds that depend on the logarithm of the number of the (meta) experts. Vovk [17] has shown that the forecasters of Herbster and Warmuth correspond to efficient implementations of the exponentially weighted forecaster run over the set of meta experts with a specific choice of the initial weights. We also refer to Auer and Warmuth [1], Bousquet and Warmuth [3], Herbster and Warmuth [9], for various extensions and powerful variants of the problem.

Another class of problems that has been investigated is when, even though no "tracking" is performed, the class of experts is very large and has a certain structure. Examples of structured classes of experts for which efficient algorithms have been constructed include prunings of decision trees (Helmbold and Schapire [5], Pereira and Singer [12]), and planar decision graphs (Takimoto and Warmuth [13]), as well as scalar quantizers for lossy data compression (György, Linder, and Lugosi [6]). These algorithms are all based on efficient implementations of the exponentially weighted average predictor. A different approach was taken by Kalai and Vempala [10] who consider Hannan's original predictor and show that it may be used to obtain efficient algorithms for a large class of problems that they call "geometric experts."

The purpose of this paper is to develop efficient algorithms to track the best expert in the case when the class of "base" experts is already very large and has some structure. Thus, in a sense, we consider a combination of the two types of problems described above. Our approach is based on a suitable modification of the original tracking algorithm of Herbster and Warmuth that allows one to apply it in the case of large, structured expert classes for which there exist efficient implementations of the exponentially weighted average prediction method. This modification is described in Section 2. In Section 3 we illustrate the method on a problem in which a base expert is associated with a path in a directed graph and the loss of a base expert is the sum of the weights over the path (that may change in every time instant). Another application involves "tracking the best quantizer" in lossy zero-delay data compression which we describe elsewhere. We also indicate how the method may be generalized to handle the tracking of general geometric experts.

2 Tracking the best expert: a variation

The aim of this section is to modify the prediction algorithm of Herbster and Warmuth [8] for tracking the best expert to allow efficient implementation if the number of experts is very large. In order to handle cases in which the set of experts is not convex, we consider randomized prediction algorithms.

The online prediction problem considered in this paper is described as follows. Suppose we want to predict the sequence $y_1, \ldots, y_T$ taking values in the set $\mathcal{Y}$ of outcomes using a sequential prediction scheme. We assume that the predictor has access to a sequence $U_1, \ldots, U_T$ of independent random variables distributed uniformly over the interval $[0, 1]$. At each time instant $t = 1, \ldots, T$, the predictor observes U_t, and based on U_t and the past input values $y^{t-1} = (y_1, \ldots, y_{t-1})$ produces an "estimate" $\hat{y}_t \in \hat{\mathcal{Y}}$ of y_t, where $\hat{\mathcal{Y}}$ is the set of predictor actions that may not be the same as $\mathcal{Y}$. Then the predictor can observe the next input symbol y_t. For simplicity we assume throughout that the total number of rounds T is fixed and known to the predictor in advance.

Formally, the prediction game is defined as follows:

Parameters: number N of base experts, outcome space $\mathcal{Y}$, action space $\hat{\mathcal{Y}}$, loss function $\ell : \mathcal{Y} \times \hat{\mathcal{Y}} \to [0, 1]$, number T of rounds.

For each round $t = 1, \ldots, T$,

(1) each (base) expert forms its prediction $f_{i,t} \in \hat{\mathcal{Y}}$, $i = 1, \ldots, N$;
(2) the forecaster observes the predictions of the base experts and the random variable U_t and chooses an estimate $\hat{y}_t \in \hat{\mathcal{Y}}$;
(3) the environment reveals the next outcome $y_t \in \mathcal{Y}$.

The *cumulative loss* of the sequential scheme at time T is given by

$$L_T = \sum_{t=1}^{T} \ell(y_t, \hat{y}_t) \ .$$

The goal of the predictor is to achieve a cumulative loss (almost) as small as the best tracking of the N base experts. More precisely, to describe the loss the predictor is compared to, consider the following "m-partition" prediction scheme: The sequence of examples is partitioned into $m+1$ contiguous segments, and on each segment the scheme assigns exactly one of the N base experts. Formally, an m-partition $\mathcal{P}(T, m, \mathbf{t}, \mathbf{e})$ of the T samples is given by an m-tuple $\mathbf{t} = (t_1, \ldots, t_m)$ such that $t_0 = 1 < t_1 < \cdots < t_m < T + 1 = t_{m+1}$, and an $(m + 1)$-vector $\mathbf{e} = (e_0, \ldots, e_m)$ where $e_i \in \{1, \ldots, N\}$. At each time instant t, $t_i \le t < t_{i+1}$, expert e_i is used to predict y_t. The cumulative loss of a partition $\mathcal{P}_{T,m,\mathbf{t},\mathbf{e}}$ is

$$L(\mathcal{P}(T, m, \mathbf{t}, \mathbf{e})) = \sum_{i=0}^{m} \sum_{t=t_i}^{t_{i+1}-1} \ell(y_t, f_{e_i,t}) = \sum_{i=0}^{m} L([t_i, t_{i+1} - 1], e_i)$$

where for any time interval I, $L(I,i) = \sum_{t\in I} \ell(y_t, f_{i,t})$ denotes the cumulative loss of expert i in I. Here and later in the paper we adopt the convention that in case a summation is empty, we define the sum to be zero (e.g., for $a > b$, $L([a,b],i) = 0$ by definition).

The goal of the predictor is to perform as well as the best partition, that is, to keep the normalized regret

$$\frac{1}{T}\left(L_T - \min_{\mathbf{t},\mathbf{e}} L(\mathcal{P}(T,m,\mathbf{t},\mathbf{e}))\right)$$

as small as possible (with high probability) for all possible outcome sequences.

Next we present a variation of the "fixed-share" share update algorithm of Herbster and Warmuth [8].

Algorithm 1 *Fix the positive numbers* η *and* $\alpha < 1$*, and initialize weights* $w^s_{1,i} = 1/N$ *for* $i = 1,\ldots,N$*. At time instants* $t = 1,2,\ldots,T$ *let* $v_t^{(i)} = w^s_{t,i}/W_t$ *where* $W_t = \sum_{i=1}^N w^s_{t,i}$*, and predict* $\hat{y}_t$ *randomly according to the distribution*

$$\mathbb{P}\{\hat{y}_t = f_{i,t}\} = v_t^{(i)}. \tag{1}$$

After observing y_t*, for all* $i = 1,\ldots,N$*, let*

$$w^m_{t,i} = w^s_{t,i} e^{-\eta\ell(y_t,f_{i,t})} \tag{2}$$

and

$$w^s_{t+1,i} = \frac{\alpha W_{t+1}}{N} + (1-\alpha) w^m_{t,i} \tag{3}$$

where $W_{t+1} = \sum_{i=1}^N w^m_{t,i}$.

Observe that $\sum_{i=1}^N w^s_{t+1,i} = \sum_{i=1}^N w^m_{t,i} = W_{t+1}$, thus there is no ambiguity in the definition of W_{t+1}. Note that equation (3) is slightly changed compared to the original algorithm of [8].

First we present a bound on the loss of the algorithm. The proof is a straightforward adaptation of the proof of [8] and therefore it is omitted.

Theorem 1. *For any positive integers* m, T*, real numbers* $0 < \alpha < 1$*,* $\eta > 0$*, and* $\delta \in (0,1)$*, and for any sequence* $y_1,\ldots,y_T$ *taking values from* $[0,1]$*, with probability at least* $1-\delta$*, the regret* L_T *of Algorithm 1 can be bounded as*

$$L_T - \min_{\mathbf{t},\mathbf{e}} L(\mathcal{P}(T,m,\mathbf{t},\mathbf{e})) \le \frac{1}{\eta}\ln\left(\frac{N^{m+1}}{\alpha^m(1-\alpha)^{T-m-1}}\right) + \frac{T\eta}{8} + \sqrt{\frac{T\ln(1/\delta)}{2}}\ . \tag{4}$$

In particular, if $\alpha = \frac{m}{T-1}$ *and* $\eta = \sqrt{8 \ln \left(\frac{N^{m+1}}{\alpha^m (1-\alpha)^{T-m-1}} \right) / T}$ *is chosen to minimize the above bound, we have*

$$\begin{aligned} &L_T - \min_{\mathbf{t},\mathbf{e}} L(\mathcal{P}(T,m,\mathbf{t},\mathbf{e})) \\ &\quad \le \sqrt{\frac{T}{2}} \sqrt{(m+1)\ln N + m \ln \frac{T-1}{m} + m} + \sqrt{\frac{T \ln(1/\delta)}{2}} \,. \end{aligned} \tag{5}$$

Remark. If the number of experts N is proportional to T^γ for some $\gamma > 0$, then, for any fixed $\delta > 0$, the bound in (5) is of order $\sqrt{(mT)\ln T}$ for large T, and so the normalized regret is

$$\frac{1}{T}\left(L_T - \min_{\mathbf{t},\mathbf{e}} L(\mathcal{P}(T,m,\mathbf{t},\mathbf{e})) \right) = O\left(\sqrt{(m/T)\ln T} \right)$$

with probability at least $1-\delta$. That is, the rate of convergence is the same (up to a constant factor) as if we competed with the best static expert on a segment of average length.

2.1 Implementation of Algorithm 1

If the number of experts N is large, for example, $N = T^\gamma$ for some large $\gamma > 1$, then the implementation of Algorithm 1 may become computationally very hard. As it is mentioned in the introduction, for several large classes of (base) experts, efficient algorithms are known to compute the exponentially weighted average predictor when no tracking is performed. The purpose of this section is to show that, whenever such an efficient algorithm is available, the tracking forecaster can also be computed efficiently by implementing Algorithm 1 in a computationally feasible way.

The main step to this direction is an alternative expression of the weights in Algorithm 1.

Lemma 1. *For any* $t = 2, \ldots, T$*, the probability* $v_t^{(i)}$ *and the corresponding normalization factor* W_t *can be obtained as*

$$v_t^{(i)} = \frac{(1-\alpha)^{t-1}}{NW_t} e^{-\eta L([1,t-1],i)} + \frac{\alpha}{NW_t} \sum_{t'=2}^{t-1} (1-\alpha)^{t-t'} W_{t'} e^{-\eta L([t',t-1],i)} + \frac{\alpha}{N} \tag{6}$$

$$W_t = \frac{\alpha}{N} \sum_{t'=2}^{t-1} (1-\alpha)^{t-1-t'} W_{t'} Z_{t',t-1} + \frac{(1-\alpha)^{t-2}}{N} Z_{1,t-1} \tag{7}$$

where $Z_{t',t-1} = \sum_{i=1}^N e^{-\eta L([t',t-1],i)}$ *is the sum of the (unnormalized) weights assigned to the experts by the exponentially weighted prediction method for the input samples* $(y_{t'}, \ldots, y_{t-1})$.

Proof. The expressions in the lemma follow directly from the recursive definition of the weights $\{w^s_{t,i}\}$. First we show that for $t = 1, \ldots, T$,

$$w^m_{t,i} = \frac{\alpha}{N} \sum_{t'=2}^{t} (1-\alpha)^{t-t'} W_{t'} e^{-\eta L([t',t],i)} + \frac{(1-\alpha)^{t-1}}{N} e^{-\eta L([1,t],i)} \tag{8}$$

$$w^s_{t+1,i} = \frac{\alpha}{N} W_{t+1} + \frac{\alpha}{N} \sum_{t'=2}^{t} (1-\alpha)^{t+1-t'} W_{t'} e^{-\eta L([t',t],i)} + \frac{(1-\alpha)^{t}}{N} e^{-\eta L([1,t],i)}. \tag{9}$$

Clearly, for a given t, (8) implies (9) by the definition (3). Since $w^s_{1,i} = 1/N$ for every expert i, (8) and (9) hold for $t = 1$ and $t = 2$ (for $t = 1$ the summations are 0 in both equations). Now assume that they hold for some $t \geq 2$. We show that then (8) holds for $t + 1$. By definition,

$$\begin{aligned}
w^m_{t+1,i} &= w^s_{t+1,i} e^{-\eta \ell(y_{t+1}, f_{i,t+1})} \\
&= \frac{\alpha}{N} W_{t+1} e^{-\eta \ell(y_{t+1}, f_{i,t+1})} + \frac{\alpha}{N} \sum_{t'=2}^{t} (1-\alpha)^{t+1-t'} W_{t'} e^{-\eta L([t',t+1],i)} \\
&\quad + \frac{(1-\alpha)^{t}}{N} e^{-\eta L([1,t+1],i)} \\
&= \frac{\alpha}{N} \sum_{t'=2}^{t+1} (1-\alpha)^{t+1-t'} W_{t'} e^{-\eta L([t',t+1],i)} + \frac{(1-\alpha)^{t}}{N} e^{-\eta L([1,t+1],i)}
\end{aligned}$$

thus (8) and (9) hold for all $t = 1, \ldots, T$. Now (6) follows from (9) by normalization for $t = 2, \ldots, T + 1$. Finally, (7) can easily be proved from (8), as for any $t = 2, \ldots, T$,

$$\begin{aligned}
W_t &= \sum_{i=1}^{N} w^m_{t-1,i} \\
&= \sum_{i=1}^{N} \left(\frac{\alpha}{N} \sum_{t'=2}^{t-1} (1-\alpha)^{t-1-t'} W_{t'} e^{-\eta L([t',t-1],i)} + \frac{(1-\alpha)^{t-2}}{N} e^{-\eta L([1,t-1],i)} \right) \\
&= \frac{\alpha}{N} \sum_{t'=2}^{t-1} (1-\alpha)^{t-1-t'} W_{t'} \sum_{i=1}^{N} e^{-\eta L([t',t-1],i)} + \frac{(1-\alpha)^{t-2}}{N} \sum_{i=1}^{N} e^{-\eta L([1,t-1],i)} \\
&= \frac{\alpha}{N} \sum_{t'=2}^{t-1} (1-\alpha)^{t-1-t'} W_{t'} Z_{t',t-1} + \frac{(1-\alpha)^{t-2}}{N} Z_{1,t-1}.
\end{aligned}$$

$\square$

Examining formula (6), one can see that the t'-th term in the summation (including the first and last individual terms) is some multiple of $e^{-\eta L([t',t-1],i)}$. The latter expression is the weight assigned to expert i by the exponentially weighted prediction method for the last $t - t'$ samples of the sequence, that is, for $(y_{t'}, \ldots, y_{t-1})$ (the last term in the summation corresponds to the case where no previous samples of the sequence are taken into consideration). Therefore, for $t \geq 2$, the random choice (1) of a predictor can be performed in two steps. First we choose a random time τ_t, which specifies how many most recent samples we are going to use for the prediction. Then we choose the predictor according to the exponentially weighted prediction for these samples. Thus, $\mathbb{P}\{\tau_t = t'\}$ is the sum of the t'-th terms with respect to the index i in the expressions for $v_t^{(i)}$, and given $\tau_t = t'$, the probability that $\hat{y}_t = f_{i,t}$ is just the probability assigned to expert i using the exponentially weighted average prediction based on the samples $(y_{t'}, \ldots, y_{t-1})$. Hence we obtain the following algorithm.

Algorithm 2 *For $t = 1$, choose $\hat{y}_1$ uniformly from the set $\{f_{1,1}, \ldots, f_{N,1}\}$. For $t \geq 2$, choose τ_t randomly according to the distribution*

$$\mathbb{P}\{\tau_t = t'\} = \begin{cases} \frac{(1-\alpha)^{t-1} Z_{1,t-1}}{N W_t} & \text{for } t' = 1 \\ \frac{\alpha(1-\alpha)^{t-t'} W_{t'} Z_{t',t-1}}{N W_t} & \text{for } t' = 2, \ldots, t \end{cases} \tag{10}$$

where we define $Z_{t,t-1} = N$. Given $\tau_t = t'$, choose $\hat{y}_t$ randomly according to the probabilities

$$\mathbb{P}\{\hat{y}_t = f_{i,t} | \tau_t = t'\} = \begin{cases} \frac{e^{-\eta L([t',t-1],i)}}{Z_{t',t-1}} & \text{for } t' = 1, \ldots, t-1 \\ \frac{1}{N} & \text{for } t' = t \end{cases} \tag{11}$$

The discussion preceding the algorithm shows that Algorithm 2 provides an alternative implementation of Algorithm 1.

Theorem 2. *Algorithm 1 and Algorithm 2 are equivalent in the sense that the generated predictor sequences have the same distribution. In particular, the sequence $(\hat{y}_1, \ldots, \hat{y}_T)$ generated by Algorithm 2 satisfies*

$$\mathbb{P}\{\hat{y}_t = f_{i,t}\} = v_t^{(i)} \tag{12}$$

for all t and i, where $v_t^{(i)}$ are the normalized weights generated by Algorithm 1.

It is not immediately obvious why Algorithm 2 is more efficient than Algorithm 1. However, in many cases the probabilities $\mathbb{P}\{\hat{y}_t = f_{i,t} | \tau_t = t'\}$ and normalization factors $Z_{t',t-1}$ may be computed efficiently, and in all those cases, since W_t can be obtained via the recursion formula (7), Algorithm 2 becomes feasible.

We need the following assumptions: For a given set of N (base) experts,

(a) the exponentially weighted average prediction method can be implemented in $O(g(T))$ time, that is, for time instants $t = 1, \ldots, T$, predictions $\hat{y}_1, \ldots, \hat{y}_T$ can be chosen sequentially according to the probabilities $\mathbb{P}\{\hat{y}_t = f_{i,t}\} = e^{-\eta L([1,t-1],i)}$ in $O(g(T))$ time for any $\eta > 0$;

(b) the sums of the weights $Z_{t-1} = \sum_{i=1}^{N} e^{-\eta L([1,t-1],i)}$ can be computed in $O(g(T))$ time for $t = 1, \ldots, T$.

Note that condition (b) is implied by the following two natural assumptions, which are often satisfied as byproducts of the efficient implementation of the exponentially weighted prediction method according to (a): for $t = 1, \ldots, T$,

(c_1) $\mathbb{P}\{\hat{y}_t = f_{i_t,t}\}$ can be computed for the chosen expert i_t (that is, $\hat{y}_t = f_{i_t,t}$) in $O(g(T))$ time;

(c_2) the cumulative losses $L([1, t-1], i_t)$ of the chosen experts i_t can be computed in $O(g(T))$ time.

Then Z_{t-1} can be calculated as $Z_{t-1} = e^{-\eta L([1,t-1],i_t)} / \mathbb{P}\{\hat{y}_t = f_{i_t,t}\}$.

The next theorem shows that, under assumptions (a) and (b) on the class of the base experts, Algorithm 2 can be implemented efficiently, and thus tracking can be performed with low computational complexity.

Theorem 3. *Assume that for the set of base experts conditions (a) and (b) are satisfied. Then Algorithm 2 can be implemented in* $O\left(T^2 + \sum_{t=1}^{T} g(t)\right)$ *time for* T *rounds.*

Proof. For $t = 1$ choose $\hat{y}_1$ uniformly from $\{f_{1,1}, \ldots, f_{N,1}\}$, and set $W_1 = 1$. For each $t = 2, \ldots, T$, run the exponentially weighted prediction algorithm for the base experts with the reverse set of examples $y_{t-1}, \ldots, y_1$ as input data and compute the constants $Z_{t',t-1}$ for all $t' = 1, \ldots, t-1$ in $O(g(t))$ time. Then compute W_t from $Z_{1,t-1}, \ldots, Z_{t,t-1}$ (recall that $Z_{t,t-1} = N$) and $W_1, \ldots, W_{t-1}$ according to (7) in $O(t)$ time. Then the choice of τ_t according to (10) can be performed in $O(t)$ time, and the prediction according to (11) can be chosen in $O(g(t))$ time. Thus, at time instant t, $O(g(t)) + O(t)$ computations are required, giving overall computational complexity $O(T^2 + \sum_{t=1}^{T} g(t))$. $\square$

We illustrate the use of this algorithm in just one special case when the losses of the base experts are given by weights of a path in a directed graph. This application, that is, in a sense, a generic example, should serve as an illustration. In the full version of the paper other examples will be given.

3 Minimum weight path in a directed graph

In this section we present an application of Algorithm 2 where the constants $Z_{t',t}$ can be computed efficiently as discussed at the end of the previous section. We consider the problem of tracking the minimum-weight path of a given length in a weighted directed graph. Other efficient implementations of exponentially weighted prediction methods, such as for finding the minimum weight path (of unrestricted length) in a weighted directed acyclic graph in Takimoto and Warmuth [14],[15], can also be combined with our tracking method in a similar way.

Formally, we have a directed graph $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the set of nodes and edges, respectively. Given a fixed pair of nodes s and u, let $\mathcal{R}_M$ denote the set of all directed paths of length M from s to u, let $N = |\mathcal{R}_M|$ denote the number of such paths, and assume that $\mathcal{R}_M$ is not empty (that is, $N > 0$). We also assume that for all $z \neq u$, $z \in \mathcal{V}$, there is an edge starting from z. (Otherwise node z is of no use in finding a path from s to u, and all such nodes can be removed from the graph at the beginning of the algorithm in $O(|\mathcal{V}|)+O(|\mathcal{E}|)$ time, parallel with reading the description of the graph.) At time instants $t = 1, 2, \ldots$ the predictor picks a path $\hat{y}_t \in \mathcal{R}_M$. The cost of this path is the sum of the weights $\delta_t(a)$ on the edges a of the path (the weights are assumed to be nonnegative real numbers), which are revealed for each $a \in \mathcal{E}$ only after the path has been chosen. To use our previous definition for prediction, we may define $y_t = \{\delta_t(a)\}_{a \in \mathcal{E}}$, and the loss function as

$$\ell(y_t, \hat{y}_t) = \sum_{a \in \hat{y}_t} \delta_t(a)$$

for each pair $(y_t, \hat{y}_t)$. The cumulative loss at time instant T is given as

$$L_T = \sum_{t=1}^{T} \ell(y_t, \hat{y}_t).$$

Our goal is to perform as well as the best combination of paths (base experts) which is allowed to change the path m times during time instants $t = 1, \ldots, T$. As in the prediction context, such a combination is given as an m-partition $\mathcal{P}(T, m, \mathbf{t}, \mathbf{e})$, where $\mathbf{t} = (t_1, \ldots, t_m)$ such that $t_0 = 1 < t_1 < \cdots < t_m < t_{m+1} = T + 1$, and $\mathbf{e} = (e_0, \ldots, e_m)$, where $e_i \in \mathcal{R}_M$ (that is, expert $e \in \mathcal{R}_M$ predicts $f_{e,t} = e$). The cumulative loss of a partition $\mathcal{P}(T, m, \mathbf{t}, \mathbf{e})$ is

$$L(\mathcal{P}(T, m, \mathbf{t}, \mathbf{e})) = \sum_{i=0}^{m} \sum_{t=t_i}^{t_{i+1}-1} \ell(y_t, e_i) = \sum_{i=0}^{m} \sum_{t=t_i}^{t_{i+1}-1} \sum_{a \in e_i} \delta_t(a).$$

Now Algorithms 1 and 2 can be used to choose the path $\hat{y}_t$ randomly at each time instant $t = 1, \ldots, T$, and the regret

$$L_T - \min_{\mathbf{t}, \mathbf{e}} L(\mathcal{P}(T, m, \mathbf{t}, \mathbf{e}))$$

can be bounded by Theorem 1. The question is whether in this setup we can compute efficiently a path based on the exponentially weighted prediction method and the constants $Z_{t',t}$. The following theorem gives a positive answer.

Theorem 4. *For the minimum weight path problem described in this section, Algorithm 2 can be implemented in $O(T^2 M|\mathcal{E}|)$ time. If $\alpha = m/(T-1)$, $\delta_t(a) <$*

$1/M$ for all time instants t and edges $a \in \mathcal{E}$, and $\eta = \sqrt{8 \ln \left(\frac{N^{m+1}}{\alpha^m (1-\alpha)^{T-m-1}} \right) / T}$, then the regret of the algorithm can be bounded from above, with probability at least $1-\delta$, as

$$
\begin{aligned}
&L_T - \min_{\mathbf{t},\mathbf{e}} L(\mathcal{P}(T, m, \mathbf{t}, \mathbf{e})) \\
&\qquad \le \sqrt{\frac{T}{2}} \sqrt{(m+1)\ln N + m \ln \frac{T-1}{m} + m} + \sqrt{\frac{T \ln(1/\delta)}{2}} .
\end{aligned}
$$

Proof. The bound in the theorem follows trivially from the optimized bound (5) in Theorem 1. All we need to show is that the algorithm can be implemented in $O(T^2 M|\mathcal{E}|)$ time. To do this, we show that the exponentially weighted average prediction method for T rounds can be implemented in $O(TM|\mathcal{E}|)$ time for the above described minimum weight path problem. Then the result follows by Theorem 3. In the following we modify the algorithm of György, Linder, and Lugosi [6] to choose a path $\hat{y}_t$ randomly based on $(y_1, y_2, \ldots, y_{t-1})$ (that is, based on the weights $\{\delta_j(a)\}_{a\in\mathcal{E}}, j \in [1, t-1]$) according to the probabilities

$$
\mathbb{P}\{\hat{y}_t = r\} = \frac{e^{-\eta \sum_{a\in r} \Delta_{t-1}(a)}}{\sum_{r' \in \mathcal{R}_M} e^{-\eta \sum_{a \in r'} \Delta_{t-1}(a)}} \tag{13}
$$

where $\Delta_{t-1}(a) = \sum_{j=1}^{t-1} \delta_j(a)$, and compute

$$
Z_{t-1} = \sum_{r \in \mathcal{R}_M} e^{-\eta \sum_{a \in r} \Delta_{t-1}(a)}.
$$

We show that for $t = 1, \ldots, T$, this can be done in $O(TM|\mathcal{E}|)$ time, yielding that the problem satisfies conditions (a) and (b) with $g(T) = TM|\mathcal{E}|$.

For any $z \in \mathcal{V}$ and $k = 1, \ldots, M$, let $\mathcal{R}_k^z$ denote the set of paths of length k from z to u, and let $G_{t-1}(z,k)$ denote the sum of the exponential cumulative losses in the interval $[1, t-1]$ of all paths in $\mathcal{R}_k^z$. Formally, if $\mathcal{R}_k^z$ is empty then we define $G_{t-1}(z,k) = 0$, otherwise

$$
G_{t-1}(z,k) = \sum_{r \in \mathcal{R}_k^z} e^{-\eta \sum_{a\in r} \Delta_{t',t-1}(a)}. \tag{14}
$$

The function $G_{t-1}(z,k)$ will prove useful in computing Z_{t-1}, as $Z_{t-1} = G_{t-1}(s, M)$, and in drawing $\hat{y}_t$ randomly for a given τ_t: Instead of computing the cumulative losses $\sum_{a \in r} \Delta_{t-1}(a)$ for all $r \in \mathcal{R}_M$ (needed by (13)), following the algorithm of [6], we can draw the path $\hat{y}_t$ by drawing its edges successively. Denote the jth node along a path $r \in \mathcal{R}_M$ by $z_{r,j}$ for $j = 0, \ldots, M$, where $z_{r,0} = s$ and $z_{r,M} = u$. Then, for any $k = 1, \ldots, M-1$, the probability that the kth node in the path $\hat{y}_t$ is z_k given that the previous nodes are $z_0, z_1, \dots, z_{k-1}$ is given by

$$
\begin{aligned}
&\mathbb{P}\{z_{\hat{y}_t,k} = z_k | z_{\hat{y}_t,j} = z_j, j = 0, \dots, k-1\} \\
&\qquad = \frac{\mathbb{P}\{z_{\hat{y}_t,j} = z_j, j = 0, \dots, k\}}{\mathbb{P}\{z_{\hat{y}_t,j} = z_j, j = 0, \dots, k-1\}} \\
&\qquad = \frac{\sum_{r: z_{r,i} = z_i, i=0,\dots,k} e^{-\eta \sum_{j=1}^{M} \Delta_{t-1}((z_{r,j-1}, z_{r,j}))}}{\sum_{r: z_{r,i} = z_i, i=0,\dots,k-1} e^{-\eta \sum_{j=1}^{M} \Delta_{t-1}((z_{r,j-1}, z_{r,j}))}} \\
&\qquad = e^{-\eta \Delta_{t-1}((z_{k-1}, z_k))} \frac{G_{t-1}(z_k, M-k)}{G_{t-1}(z_{k-1}, M-k+1)}. \qquad (15)
\end{aligned}
$$

Therefore, given the functions Δ_{t-1} and G_{t-1}, $\hat{y}_t$ and its probability can be computed in $O(M|\mathcal{V}|)$ steps using the exponentially weighted average prediction method.

Next we show how to compute G_{t-1}. For any node $z \in \mathcal{V}$, let $\mathcal{E}(z)$ denote the set of edges starting at z. As any path of length $k \geq 2$ can be decomposed as the first edge in the path and the remaining path of length $k-1$, it is easy to see that for any $M \geq k \geq 2$, $G_{t-1}(z, k)$ can be computed recursively as

$$
G_{t-1}(z, k) = \sum_{\hat{z}: (z, \hat{z}) \in \mathcal{E}(z)} G_{t-1}(\hat{z}, k-1) e^{-\eta \Delta_{t-1}((z, \hat{z}))} \qquad (16)
$$

and

$$
G_{t-1}(z, 1) = \begin{cases} e^{-\eta \Delta_{t-1}((z,u))} & \text{if } (z, u) \in \mathcal{E}; \\ 0 & \text{otherwise.} \end{cases}
$$

When calculating (16) for a given k, each edge is taken into consideration exactly once (and we have to do the update of G for each node). Thus, assuming that the cumulative weights $\Delta_{t-1}(a)$ are known for each edge $a \in \mathcal{E}$, the computational cost of calculating $G_{t-1}(z, k)$ for a given k is $O(|\mathcal{E}|) + O(|\mathcal{V}|) = O(|\mathcal{E}|)$ (as by assumption, $|\mathcal{E}| \geq |\mathcal{V}| - 1$). Therefore, the computational complexity of calculating $G_{t-1}(z, k)$ for all z and k, given the cumulative weights $\Delta_{t-1}(a)$ are known, is $O(M|\mathcal{E}|)$. Now as t increases from 1 to T, if we store the cumulative weights $\Delta_{t-1}(a)$ for each edge a, then only $O(|\mathcal{E}|)$ computations are needed to update the cumulative weights at the edges for each value of t. Therefore, calculating $G_{t-1}(z, k)$ for all $z \in \mathcal{V}$, $1 \leq k \leq M$, and $t = 1, \dots, T$ requires $O(TM|\mathcal{E}|)$ computations. This shows that conditions (a) and (b) are satisfied for this problem with $g(T) = TM|\mathcal{E}|$. Applying Theorem 3 finishes the proof. □

Remarks.

(i) If we assume that the graph contains no cycle with a negative weight at any time instant, then the minimum weight path (of unrestricted length) is of length at most $|\mathcal{V} - 1|$. Therefore, the algorithm can easily be modified to compete with paths of unrestricted length. All we require is an additional cycle in which M goes from 1 to $|\mathcal{V}| - 1$ to examine all possible paths. Then, in the random choice of the path, after choosing τ_t, we randomly decide the length of the path and choose a path of that length using exponential weighting. The bound on the regret remains the same as in Theorem 4; the price we pay is an increase in the complexity of the algorithm which becomes $O(T^2|\mathcal{V}|^2|\mathcal{E}|)$.

(ii) If the graph is acyclic, then the above algorithm can be simplified as there is no need to keep track the second parameter of the function G_{t-1} (this is basically an application of the weight pushing algorithm of Takimoto and Warmuth [14],[15] to the graph for the time interval $[1, t-1]$). Then the minimum weight path (of unrestricted length) can be tracked in $O(T^2|\mathcal{E}|)$ time, while the bound on the regret still holds.

(iii) It is also possible to apply the above algorithm for tracking the best geometric expert. A geometric expert is a combination of "sub-experts" from a given set, such that the loss of a geometric expert equals the sum of the losses of its "sub-experts"; however, not all possible combinations of the "sub-experts" are allowed (for a formal definition of the problem, see Kalai and Vempala [10]). An example of the geometric expert problem is the minimum weight path problem in a graph, where the "sub-experts" are the edges and the allowed geometric (combined) experts are the paths. However, the geometric expert problem can also be treated as a special case of the minimum weight path problem, as one can easily construct a graph such that there is a one-to-one correspondence between paths of the graph (between to given nodes) and the allowed geometric experts: each edge of the graph corresponds to a "sub-expert", and each path corresponds to the geometric expert combined from the "sub-experts" corresponding to its edges. Note that usually several edges correspond to each "sub-expert". In this way it is possible to track the best geometric expert using the graph algorithms of this section. However, the complexity of the algorithm depends heavily on the number of edges of the graph, and it is not clear at all how one can create a graph with a minimum number of edges for a given set of geometric experts.

References

1. P. Auer and M.K. Warmuth. Tracking the best disjunction. *Machine Learning*, 32(2):127–150, 1998.
2. D. Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6:1–8, 1956.
3. O. Bousquet and M. K. Warmuth. Tracking a small set of experts by mixing past posteriors. *Journal of Machine Learning Research*, 3:363–396, Nov. 2002.
4. N. Cesa-Bianchi, Y. Freund, D. P. Helmbold, D. Haussler, R. Schapire, and M. K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.
5. R.E. Schapire D.P. Helmbold. Predicting nearly as well as the best pruning of a decision tree. *Machine Learning*, 27:51–68, 1997.
6. A. György, T. Linder, and G. Lugosi. Efficient algorithms and minimax bounds for zero-delay lossy source coding. *IEEE Transactions on Signal Processing*, pages 2337–2347, Aug. 2004.
7. J. Hannan. Approximation to Bayes risk in repeated plays. In M. Dresher, A. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, volume 3, pages 97–139. Princeton University Press, 1957.
8. M. Herbster and M. K. Warmuth. Tracking the best expert. *Machine Learning*, pages 1–29, 1998.

9. M. Herbster and M.K. Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1:281–309, 2001.
10. A. Kalai and S. Vempala. Efficient algorithms for online decision problems. In B. Schölkopf and M. K. Warmuth, editors, *COLT 2003*, LNAI 2777, pages 26–40, Berlin–Heidelberg, 2003. Springer-Verlag.
11. N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212–261, 1994.
12. F. Pereira and Y. Singer. An efficient extension to mixture techniques for prediction and decision trees. *Machine Learning*, 36:183–199, 1999.
13. E. Takimoto and M. Warmuth. Predicting nearly as well as the best pruning of a planar decision graph. *Theoretical Computer Science*, 288:217–235, 2002.
14. E. Takimoto and M. K. Warmuth. Path kernels and multiplicative updates. In J. Kivinen and R. H. Sloan, editors, *COLT 2002*, LNAI 2375, pages 74–89, Berlin–Heidelberg, 2002. Springer-Verlag.
15. E. Takimoto and M. K. Warmuth, "Path kernels and multiplicative updates," *Journal of Machine Learning Research*, vol. 4, pages 773–818, 2003.
16. V. Vovk. Aggregating strategies. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 372–383, New York, 1990. Association of Computing Machinery.
17. V. Vovk. Derandomizing stochastic prediction strategies. *Machine Learning*, 35(3):247–282, 1999.