A NOTE ON PARAMETER TUNING FOR ON-LINE SHIFTING ALGORITHMS

OLIVIER BOUSQUET

ABSTRACT. In this short note, building on ideas of M. Herbster [2] we propose a method for automatically tuning the parameter of the FIXED-SHARE algorithm proposed by Herbster and Warmuth [3] in the context of on-line learning with shifting experts. We show that this can be done with a memory requirement of $O(nT)$ and that the additional loss incurred by the tuning is the same as the loss incurred for estimating the parameter of a Bernoulli random variable.

1. SETTING

How setting is the same as in [3]. We consider n experts and at each time period $t = 1, \ldots, T$ they make predictions and incur a loss $L(t, i)$ (i being the index of the expert) which we model here as a negative log-likelihood, so that the probability that expert i makes a correct prediction at time t is $e^{-L(t,i)}$. We consider a Bayesian setting which we will use to motivate the updates.

In each step one expert is supposed to be better than the other ones (i.e. having smaller loss) and this best expert changes from time to time. This is the so-called shifting experts framework. We denote by y_t the observed outcome and by e_t the index of the best expert at time t. E_t will denote the random variable which models this index. Boldface notation corresponds to sequences, e.g. $\boldsymbol{y}_t = y_1, \ldots, y_t$.

2. FIXED α

We now model the way the best expert changes (or shifts). This is done via the following probabilistic model

$$P(e_t|\boldsymbol{e}_{t-1}) = \begin{cases} 1-\alpha & \text{if } e_t = e_{t-1} \\ \frac{\alpha}{n-1} & \text{otherwise} \end{cases}$$

with $P(e_1) = \frac{1}{n}$. This means that initially all the experts are equally likely and then at each step the next expert can be the same as the previous one with probability $1-\alpha$ and can be any other (equally likely) with probability

α. We thus get the following prior over sequences with k shifts:

$$P(\boldsymbol{e}_T) = \prod_{t=1}^{T} P(e_t|\boldsymbol{e}_{t-1}) = \frac{1}{n}(1-\alpha)^{T-k-1}\left(\frac{\alpha}{n-1}\right)^k$$

Now let us define the weights associated to the experts as(using the same notations as in [3]),

$$v_{t,i} = P(E_t = i|\boldsymbol{y}_{t-1})$$

and

$$v^m_{t,i} = P(E_t = i|\boldsymbol{y}_t)\,.$$

We easily get by the Bayes rule the so-called Loss Update,

$$v^m_{t,i} = \frac{e^{-\eta L_{t,i}} v_{t,i}}{\sum_{j=1}^n e^{-\eta L_{t,j}} v_{t,j}}\,.$$

Moreover, we have

$$P(E_{t+1} = i|\boldsymbol{y}_t) = \sum_{j=1}^{n} P(E_{t+1} = i|E_t = j, \boldsymbol{y}_t) P(E_t = j|\boldsymbol{y}_t)\,,$$

which gives the so-called Share Update

$$v_{t+1,i} = (1-\alpha)v^m_{t,i} + \frac{\alpha}{n-1}\sum_{j\neq i} v^m_{t,j}\,.$$

This shows that we recover the FIXED-SHARE algorithm of [3]. We will use this method of motivating updates to derive a tuning method for the parameter α.

3. TUNING α

Now let's consider the case where α is unknown. As usual in Bayesian approaches, we model this uncertainty by some prior distribution $P(\alpha)$ for $\alpha \in [0,1]$. The prior over possible sequences of experts is thus defined by

$$P(\boldsymbol{e}_T) = \int P(\boldsymbol{e}_T|\alpha)P(\alpha)d\alpha\,,$$

where

$$P(\boldsymbol{e}_T|\alpha) = \prod_{t=1}^{T} P(e_t|\boldsymbol{e}_{t-1}, \alpha)\,,$$

with, as before

$$P(e_t|\boldsymbol{e}_{t-1}, \alpha) = \begin{cases} 1-\alpha & \text{if } e_t = e_{t-1} \\ \frac{\alpha}{n-1} & \text{otherwise} \end{cases}\,,$$

and $P(e_1|\alpha) = \frac{1}{n}$.

For a sequence $\boldsymbol{e}_t$, let $s(\boldsymbol{e}_t)$ denote the number of shifts in that sequence, that is the number of indices t such that $e_{t+1} \neq e_t$.

We now formulate an assumption on the shape of the prior $P(\boldsymbol{e}_t)$ which actually corresponds to an assumption on the distribution $P(\alpha)$. This assumption will allow us to obtain closed-form update rules.

Assumption 1. *We assume that the prior $P(\boldsymbol{e}_T)$ can be computed recursively and depends only on the number of shifts in the sequence, that is, we assume there exist numbers $\gamma_{t,k}$ for each t and each $k < t$ such that*

$$P(\boldsymbol{e}_t) = \gamma_{t,s(\boldsymbol{e}_t)}$$

and

$$\gamma_{t,k} = (n-1)\gamma_{t+1,k+1} + \gamma_{t+1,k}$$

Notice that if $s(\boldsymbol{e}_t) = k$ we get, under the above assumption, the following relationship

$$P(e_{t+1}|\boldsymbol{e}_t) = \begin{cases} \frac{\gamma_{t+1,k}}{\gamma_{t,k}} & \text{if } e_{t+1} = e_t \\ \frac{\gamma_{t+1,k+1}}{\gamma_{t,k}} & \text{otherwise} \end{cases}$$

3.1. **Updates for Recursive Priors.** We now introduce additional vectors that will be maintained by the algorithm and derive the update rules implied by the assumption we introduced. We define for $k < t$,

$$v_{t,i,k} = P(E_t = i, s(\boldsymbol{e}_t) = k|\boldsymbol{y}_{t-1})$$

and

$$v^m_{t,i,k} = P(E_t = i, s(\boldsymbol{e}_t) = k|\boldsymbol{y}_t)$$

and as before

$$v_{t,i} = P(E_t = i|\boldsymbol{y}_{t-1})$$

$$v^m_{t,i} = P(E_t = i|\boldsymbol{y}_t)$$

We have the following result.

Theorem 1. *Under Assumption 1, the Bayes rule leads to the following algorithm. The prediction at trial t is computed using the weights*

$$v_{t,i} = \sum_{k=0}^{t-1} v_{t,i,k}$$

The Loss Updates for $k \leq t-1$ gives

$$v^m_{t,i,k} = \frac{e^{-\eta L_{t,i}} v_{t,i,k}}{\sum_{j=1}^n e^{-\eta L_{t,j}} v_{t,j}}$$

and the Share Update for $k \leq t-1$ gives

$$v_{t+1,i,k} = \frac{\gamma_{t+1,k}}{\gamma_{t,k}} v^m_{t,i,k} + \frac{\gamma_{t+1,k}}{\gamma_{t,k-1}} \sum_{j \neq i} v^m_{t,j,k-1}$$

and for $k = t$ we get

$$v_{t+1,i,t} = \frac{\gamma_{t+1,t}}{\gamma_{t,t-1}} \sum_{j \neq i} v^m_{t,j,t-1}$$

Proof. The Loss Update follows from an application of the Bayes rule

$$\begin{aligned}
&P(E_t = i, s(\boldsymbol{e}_t) = k | \boldsymbol{y}_t) \\
&= \frac{P(y_t | s(\boldsymbol{e}_t) = k, E_t = i) P(E_t = i, s(\boldsymbol{e}_t) = k | \boldsymbol{y}_{t-1})}{\sum_{j=1}^n \sum_{k=0}^{t-1} P(y_t | s(\boldsymbol{e}_t) = k, E_t = j) P(E_t = j, s(\boldsymbol{e}_t) = k | \boldsymbol{y}_{t-1})} \\
&= \frac{P(y_t | E_t = i) P(E_t = i, s(\boldsymbol{e}_t) = k | \boldsymbol{y}_{t-1})}{\sum_{j=1}^n \sum_{k=0}^{t-1} P(y_t | E_t = j) P(E_t = j, s(\boldsymbol{e}_t) = k | \boldsymbol{y}_{t-1})} \\
&= \frac{P(y_t | E_t = i) P(E_t = i, s(\boldsymbol{e}_t) = k | \boldsymbol{y}_{t-1})}{\sum_{j=1}^n P(y_t | E_t = j) P(E_t = j | \boldsymbol{y}_{t-1})}
\end{aligned}$$

For the Share Update, we write

$$\begin{aligned}
&P(E_{t+1} = i, s(\boldsymbol{e}_{t+1}) = k | \boldsymbol{y}_t) \\
&= \sum_{\boldsymbol{e}_t : s((i, \boldsymbol{e}_t)) = k} P(E_{t+1} = i, s(\boldsymbol{e}_{t+1}) = k | \boldsymbol{e}_t, \boldsymbol{y}_t) P(\boldsymbol{e}_t | \boldsymbol{y}_t) \\
&= \sum_{\boldsymbol{e}_t : s((i, \boldsymbol{e}_t)) = k} P(E_{t+1} = i | \boldsymbol{e}_t) P(\boldsymbol{e}_t | \boldsymbol{y}_t) \\
&= \sum_{\boldsymbol{e}_t : s(\boldsymbol{e}_t) = k, e_t = i} P(E_{t+1} = i | \boldsymbol{e}_t) P(\boldsymbol{e}_t | \boldsymbol{y}_t) \\
&\quad + \sum_{\boldsymbol{e}_t : s(\boldsymbol{e}_t) = k-1, e_t \neq i} P(E_{t+1} = i | \boldsymbol{e}_t) P(\boldsymbol{e}_t | \boldsymbol{y}_t) \\
&= \sum_{\boldsymbol{e}_t : s(\boldsymbol{e}_t) = k, e_t = i} \frac{\gamma_{t+1,k}}{\gamma_{t,k}} P(\boldsymbol{e}_t | \boldsymbol{y}_t) \\
&\quad + \sum_{\boldsymbol{e}_t : s(\boldsymbol{e}_t) = k-1, e_t \neq i} \frac{\gamma_{t+1,k}}{\gamma_{t,k-1}} P(\boldsymbol{e}_t | \boldsymbol{y}_t) \\
&= \frac{\gamma_{t+1,k}}{\gamma_{t,k}} P(E_t = i, s(\boldsymbol{e}_t) = k | \boldsymbol{y}_t) \\
&\quad + \frac{\gamma_{t+1,k}}{\gamma_{t,k-1}} \sum_{j \neq i} P(E_t = j, s(\boldsymbol{e}_t) = k-1 | \boldsymbol{y}_t)
\end{aligned}$$

For the Share Update with $k = t$, we have

$$\begin{aligned}
&P(E_{t+1} = i, s(\boldsymbol{e}_{t+1}) = t|\boldsymbol{y}_t) \\
&= \sum_{\boldsymbol{e}_t : s((i,\boldsymbol{e}_t))=t} P(E_{t+1} = i, s(\boldsymbol{e}_{t+1}) = t|\boldsymbol{e}_t, \boldsymbol{y}_t) P(\boldsymbol{e}_t|\boldsymbol{y}_t) \\
&= \sum_{\boldsymbol{e}_t : s((i,\boldsymbol{e}_t))=t} P(E_{t+1} = i|\boldsymbol{e}_t) P(\boldsymbol{e}_t|\boldsymbol{y}_t) \\
&= \sum_{\boldsymbol{e}_t : s(\boldsymbol{e}_t)=t-1, e_t \neq i} P(E_{t+1} = i|\boldsymbol{e}_t) P(\boldsymbol{e}_t|\boldsymbol{y}_t) \\
&= \sum_{\boldsymbol{e}_t : s(\boldsymbol{e}_t)=t-1, e_t \neq i} \frac{\gamma_{t+1,t}}{\gamma_{t,t-1}} P(\boldsymbol{e}_t|\boldsymbol{y}_t) \\
&= \frac{\gamma_{t+1,t}}{\gamma_{t,t-1}} \sum_{j \neq i} P(E_t = j, s(\boldsymbol{e}_t) = t-1|\boldsymbol{y}_t)
\end{aligned}$$

□

The outcome of the above theorem is that it is possible to implement a meta-expert algorithm for the search of the optimal α without maintaining one weight vector for each possible value of α but we need one weight vector at each time period. This means that the storage requirement of this algorithm will grow linearly in the time.

This can be an issue in practice and it might be possible to use ideas similar to the ones in [1]

3.2. **Applications.** Now let's see some examples where Assumption 1 holds. We consider two simple and widely used priors on the parameter of a Bernoulli random variable: the uniform and the $(1/2, 1/2)$-Dirichlet priors.

Proposition 1. *If we put a uniform prior over the values of α the coefficients $\gamma_{t,k}$ are recursively defined by*

$$\gamma_{0,0} = \frac{1}{n}$$

and

$$\gamma_{t+1,k} = \frac{t-k}{t+1}\gamma_{t,k}$$

$$\gamma_{t+1,k+1} = \frac{k+1}{(n-1)(t+1)}\gamma_{t,k}$$

and the Share Update is

$$v_{t+1,i,k} = \frac{t-k}{t+1} v^m_{t,i,k} + \frac{k}{(n-1)(t+1)} \sum_{j \neq i} v^m_{t,j,k-1}$$

Proof. We have, for a sequence $\boldsymbol{e}_t$ with k shifts,

$$P(\boldsymbol{e}_t) = \int P(\boldsymbol{e}_t|\alpha)P(\alpha)d\alpha = \int_0^1 \frac{1}{n}(1-\alpha)^{t-k-1}\left(\frac{\alpha}{n-1}\right)^k d\alpha$$

The initial condition is trivial, then we have

$$\begin{aligned}(n-1)\gamma_{t+1,k} + \gamma_{t+1,k+1} &= \frac{1}{n(n-1)^k}\int_0^1 (1-\alpha)^{t-k}\alpha^k + (1-\alpha)^{t-k-1}\alpha^{k+1}d\alpha \\ &= \frac{1}{n(n-1)^k}\int_0^1 (1-\alpha)^{t-k-1}\alpha^k d\alpha = \gamma_{t,k}\end{aligned}$$

Also, by integration by parts

$$\begin{aligned}\gamma_{t+1,k} &= \frac{1}{n(n-1)^k}\int_0^1 (1-\alpha)^{t-k}\alpha^k d\alpha \\ &= \frac{t-k}{(k+1)n(n-1)^k}\int_0^1 (1-\alpha)^{t-k-1}\alpha^{k+1}d\alpha \\ &= \frac{(t-k)(n-1)}{k+1}\gamma_{t+1,k+1}\end{aligned}$$

which gives the updates by simple algebra. □

Proposition 2. *If we put a $(\frac{1}{2},\frac{1}{2})$-Dirichlet prior over the values of α the coefficients $\gamma_{t,k}$ are recursively defined by*

$$\gamma_{0,0} = \frac{1}{n}$$

and

$$\gamma_{t+1,k} = \frac{t-k-1/2}{t}\gamma_{t,k}$$
$$\gamma_{t+1,k+1} = \frac{k+1/2}{(n-1)t}\gamma_{t,k}$$

and the Share Update is

$$v_{t+1,i,k} = \frac{t-k-1/2}{t}v^m_{t,i,k} + \frac{k-1/2}{(n-1)t}\sum_{j\neq i} v^m_{t,j,k-1}$$

Proof. We have, for a sequence $\boldsymbol{e}_t$ with k shifts,

$$P(\boldsymbol{e}_t) = \int P(\boldsymbol{e}_t|\alpha)d\alpha = \frac{1}{\pi n(n-1)^k}\int_0^1 (1-\alpha)^{t-k-1-1/2}\alpha^{k-1/2}d\alpha$$

For the initial condition we write

$$\int_0^1 \frac{1}{\sqrt{(1-\alpha)\alpha}}d\alpha = \int_0^{\pi/2}\frac{1}{\sin\theta\cos\theta}d\sin^2\theta = \int_0^{\pi/2} 2d\theta = \pi$$

Then we have

$$\begin{aligned}
&(n-1)\gamma_{t+1,k} + \gamma_{t+1,k+1} \\
&= \frac{1}{\pi n(n-1)^k}\int_0^1 (1-\alpha)^{t-k-1/2}\alpha^{k-1/2} + (1-\alpha)^{t-k-1-1/2}\alpha^{k+1/2}d\alpha \\
&= \frac{1}{\pi n(n-1)^k}\int_0^1 (1-\alpha)^{t-k-1-1/2}\alpha^{k-1/2}d\alpha = \gamma_{t,k}
\end{aligned}$$

Also, by integration by parts

$$\begin{aligned}
\gamma_{t+1,k} &= \frac{1}{\pi n(n-1)^k}\int_0^1 (1-\alpha)^{t-k-1/2}\alpha^{k-1/2}d\alpha \\
&= \frac{t-k-1/2}{(k+1/2)\pi n(n-1)^k}\int_0^1 (1-\alpha)^{t-k-1-1/2}\alpha^{k+1/2}d\alpha \\
&= \frac{(t-k-1/2)(n-1)}{k+1/2}\gamma_{t+1,k+1}
\end{aligned}$$

which gives the updates by simple algebra. □

4. BOUNDS

In this section we derive bounds that generalize the results of [3] to the case where α is tuned online.

We denote by $L_{1..T,A}$ the loss incurred by the algorithm from time 1 to T and by $L_{1..T,\boldsymbol{e}_T}$ the loss incurred by an ideal algorithm which would use the sequence $\boldsymbol{e}_T$ of experts to make up its predictions. Our goal is to compare the loss of the algorithm to the loss of the best such sequence.

In [3] results are given for a fixed choice of α and it is shown that the bound is optimized when $\alpha = k/(T-1)$ where k is the number of shifts in the sequence.

We have the following theorems

Theorem 2. *With the uniform prior, the following bounds holds*

$$\begin{aligned}
L_{1..T,A} \leq\ & L_{1..T,\boldsymbol{e}_T} + c\ln n + ck\ln(n-1) \\
& + ck\ln\frac{T-1}{k} + c(T-k-1)\ln\frac{T-1}{T-k-1} \\
& + c\ln 2 + \frac{c}{2}\ln(T-1)
\end{aligned}$$

and thus the additional cost of estimating α compared to setting $\alpha = \frac{k}{T-1}$ is

$$c\ln 2 + \frac{c}{2}\ln(T-1)$$

Proof. The proof follows from the same arguments as in [3] or [1]. By the fundamental Lemma of [1] we have

$$L_{1..T,A} \leq -c \ln P(\boldsymbol{y}_T) = L_{1..T,\boldsymbol{e}_T} - c \ln P(\boldsymbol{e}_T) \,,$$

where in our case

$$\ln P(\boldsymbol{e}_T) = \ln n + k \ln(n-1) + \ln \int_0^1 (1-\alpha)^{T-k-1} \alpha^k P(\alpha) d\alpha \,,$$

which we can write

$$\ln P(\boldsymbol{e}_T) = \sum_{t=1}^T \ln P(e_t | \boldsymbol{e}_{t-1}) = \sum_{t=1}^T \ln \frac{\gamma_{t,s(e_t)}}{\gamma_{t-1,s(e_{t-1})}} = \ln \gamma_{T,s(\boldsymbol{e}_T)}$$

Using the recursion formulas, it is easy to check that

$$\gamma_{T,k} = \frac{1}{n(n-1)^k} \binom{T-1}{k}^{-1}$$

□

Theorem 3. *With the Dirichlet prior, the following bounds holds*

$$\begin{aligned} L_{1..T,A} \quad \leq \quad & L_{1..T,\boldsymbol{e}_T} + c \ln n + ck \ln(n-1) \\ & + ck \ln \frac{T-1}{k} + c(T-k-1) \ln \frac{T-1}{T-k-1} \\ & + c \ln 2 + \frac{c}{2} \ln(T-1) \end{aligned}$$

and thus the additional cost of estimating α compared to setting $\alpha = \frac{k}{T-1}$ is

$$c \ln 2 + \frac{c}{2} \ln(T-1)$$

Proof. We have

$$\frac{1}{pi} \int_0^1 (1-\alpha)^{T-k-1-1/2} \alpha^{k-1/2} d\alpha \geq \frac{1}{2} \frac{1}{\sqrt{T-1}} \left(\frac{k}{T-1} \right)^k \left(\frac{T-k-1}{T-1} \right)^{T-k-1}$$

thus

$$\gamma_{T,k} \geq \frac{1}{2n(n-1)^k} \frac{1}{\sqrt{T-1}} \left(\frac{k}{T-1} \right)^k \left(\frac{T-k-1}{T-1} \right)^{T-k-1}$$

□

Notice that the additional loss due to the adaptive tuning of α is exactly the cost of estimating the parameter of a Bernouilli sequence.

ACKNOWLEDGEMENTS

The author is grateful to Manfred Warmuth for many insights and inspiring discussions and to Mark Herbster for providing some unpublished notes on this topic which served as a starting point for this work.

REFERENCES

[1] O. Bousquet and M. Warmuth. Tracking a small set of experts by mixing past posteriors. *Journal of Machine Learning Research*, 3:363–396, 2002.
[2] M. Herbster. Tracking the best expert II. Unpublished Manuscript, 1997.
[3] M. Herbster and M. K. Warmuth. Tracking the best expert. *Journal of Machine Learning*, 32(2):151–178, August 1998. Special issue on concept drift.

MAX PLANCK INSTITUTE FOR BIOLOGICAL CYBERNETICS, SPEMANNSTR. 38, D-72076 TÜBINGEN, GERMANY, olivier.bousquet@tuebingen.mpg.de