Lecture 11

Instructor: Haipeng Luo

1 The Fixed-share Algorithm

In this lecture we come back to the expert problem and introduce a specific algorithm for non-stationary environments. We start with considering the switching regret against a sequence of experts

$$\mathcal{R}_T(i_1, \ldots, i_T) = \sum_{t=1}^{T} \langle p_t, \ell_t \rangle - \ell_t(i_t)$$

where $i_1, \ldots, i_T \in [N]$ is such that $\sum_{t=2}^{T} \mathbf{1}\{i_t \neq i_{t-1}\} = S - 1$. According to discussions from previous lectures, we know that a strongly adaptive algorithm can achieve regret of order $\mathcal{O}(\sqrt{TS\ln(NT)})$, and a strongly adaptive algorithm can be constructed through a sleeping expert algorithm.

Now we discuss a very different way to derive a simple algorithm with similar regret bounds. The idea is to cast the problem as another expert problem with a set of more complicated experts. In this new expert problem, each expert (called meta-expert) can be represented by $e \in [N]^T$, a sequence of the original experts. The set of all meta-experts is $\mathcal{M} = \{e \in [N]^T : \sum_{t=2}^{T} \mathbf{1}\{e(t) \neq e(t-1)\} = S - 1\}$, that is, sequences with $S - 1$ switches. The cardinality of this set $M = |\mathcal{M}|$ is bounded by $\binom{T-1}{S-1} N^S$. Finally the loss of meta-expert e at time t is simply $\widehat{\ell}_t(e) \overset{\text{def}}{=} \ell_t(e(t))$, the loss of the original expert $e(t)$ at time t.

Suppose we apply an expert algorithm with regular regret guarantee to this new expert problem, and let $\widehat{p}_t(e)$ be the weight on meta-expert e at time t. Note that for any $e^\star \in \mathcal{M}$, the regret against this meta-expert $e^\star$ is

$$\begin{aligned}\sum_{t=1}^{T} \left\langle \widehat{p}_t, \widehat{\ell}_t \right\rangle - \widehat{\ell}_t(e^\star) &= \sum_{t=1}^{T} \sum_{e \in \mathcal{M}} \widehat{p}_t(e) \ell_t(e(t)) - \ell_t(e^\star(t)) \\ &= \sum_{t=1}^{T} \sum_{i=1}^{N} \left(\sum_{e \in \mathcal{M}, e(t)=i} \widehat{p}_t(e) \right) \ell_t(i) - \ell_t(e^\star(t)),\end{aligned}$$

which implies that if we let $p_t(i) = \sum_{e \in \mathcal{M}, e(t)=i} \widehat{p}_t(e)$ and $e^\star$ be such that $e^\star(t) = i_t$, then the regular regret in the new expert problem is exactly the switching regret $\mathcal{R}_T(i_1, \ldots, i_T)$ in the original problem. Moreover, the former should be of order $\sqrt{T \ln M} = \sqrt{T \ln \left(\binom{T-1}{S-1} N^S \right)} = \sqrt{TS \ln \left(\frac{NT}{S} \right)}$, which is the same bound we have shown for switching regret (in fact even slightly better).

The methodology above is extremely useful in quickly establishing a regret upper bound for a complicated problem and to get a sense of what is information-theoretically possible, and often time such bound also turns out to be optimal. However, the resulting algorithm is often inefficient in terms of running time because the new expert problem has a huge set of experts. This is indeed the case in the example above where M is exponential in S. Since typical expert algorithms such as Hedge have linear (in M) running time, this reduction does not lead to an efficient algorithm.

The way to address this issue is at first glance counter-intuitive: we will actually switch to an *even larger* set of experts $\mathcal{M} = \{e \in [N]^T\}$ (that is, all the possible sequences of length T). Clearly, the cardinality $M = |\mathcal{M}|$ becomes N^T, which is exponential in T. Even ignoring efficiency issues, this seems like a terrible idea since now the regular regret is $\mathcal{O}(\sqrt{T \ln M}) = \mathcal{O}(T\sqrt{\ln N})$, which is linear in T. This is actually consistent with our lower bound for dynamic regret: without any assumptions on the competitor sequence, one should not be able to obtain sublinear regret.

However, it turns out that we can address both issues (inefficiency and linear regret) at the same time by using Hedge *with a proper prior distribution*. Recall that the Hedge prediction (using current notation) is $\widehat{p}_{t+1}(e) \propto \exp(-\eta \sum_{\tau=1}^{t} \widehat{\ell}_\tau(e))$. Although we did not discuss this, similar to Squint it is straightforward to allow a prior distribution $\widehat{p}_1$ for Hedge and rewrite the update rule as

$$\widehat{p}_{t+1}(e) \propto \widehat{p}_1(e) \exp\left(-\eta \sum_{\tau=1}^{t} \widehat{\ell}_\tau(e)\right),$$

and the regret bound against any distribution $q \in \Delta(M)$ is

$$\sum_{t=1}^{T} \left\langle \widehat{p}_t - q, \widehat{\ell}_t \right\rangle \le \frac{\mathrm{KL}(q, \widehat{p}_1)}{\eta} + T\eta \qquad (1)$$

(Try to verify why this is true. Hint: the mirror descent framework is the easiest way to show this). The key is now to pick $\widehat{p}_1$ so that the prior for meta-experts with a small number of switches is large, which then implies small $\mathrm{KL}(q, \widehat{p}_1)$ when q concentrates on such meta-experts. This motivates the following prior that is defined through a Markov process:

$$\widehat{p}_1(e) = \pi(e(1)) \prod_{t=2}^{T} \pi(e(t)|e(t-1))$$

where $\pi(i) = 1/N$ for $i \in [N]$ is the initial distribution and

$$\pi(i|j) = \begin{cases} (1-\beta) & \text{if } i = j \\ \beta/(N-1) & \text{else} \end{cases}$$

for $i, j \in [N]$ and some parameter $\beta \in [0,1]$ is the transition probability from j to i. In other words, one can imagine that each meta-expert e is created by first drawing an initial expert $e(1) \sim \pi(\cdot)$ (that is, uniformly), and then transiting to the next expert one by one with $e(t) \sim \pi(\,\cdot\,|e(t-1))$ (that is, with probability $1-\beta$ stay at the same expert, otherwise transit to one of other $N-1$ experts uniformly at random). Note that this is merely a thought experiment to define the prior $\widehat{p}_1$ – the algorithm is not actually doing this kind of sampling, nor is the environment.

It is then clear that the smaller the parameter β, the larger the probability of creating a meta-expert with a small number of switches. Specifically, the prior for e with $\sum_{t=2}^{T} \mathbf{1}\{e(t) \neq e(t-1)\} = S-1$ is at exactly $\frac{1}{N}(1-\beta)^{T-S}(\frac{\beta}{N-1})^{S-1}$. Therefore, if q concentrates on e then

$$\mathrm{KL}(q, \widehat{p}_1) = -\ln \widehat{p}_1(e) \le S \ln N + (T-S) \ln\left(\frac{1}{1-\beta}\right) + S \ln\left(\frac{1}{\beta}\right)$$

where the last two terms is minimized when $\beta = S/T$ with minimum value $T \cdot H(S/T)$ for the binary entropy function $H(\rho) = (1-\rho)\ln\frac{1}{1-\rho} + \rho \ln \frac{1}{\rho}$. One can also verify[1] the fact $H(\rho) \le \rho(1 + \ln\frac{1}{\rho})$ and therefore $\mathrm{KL}(q, \widehat{p}_1) = \mathcal{O}\left(S \ln\left(\frac{NT}{S}\right)\right)$ Plugging this into Eq. (1) we again obtain switching regret $\mathcal{O}\left(\sqrt{TS \ln\left(\frac{NT}{S}\right)}\right)$.

The discussion above shows that the specific prior addresses the linear regret issue. Next we show how it also allows efficient implementation. Indeed, keep in mind that ultimately we only care about getting $p_t(i) = \sum_{e:e(t)=i} \widehat{p}_t(e)$ but only the individual $\widehat{p}_t(e)$. To compute $p_t(i)$, first note that

$$p_{t+1}(i) = \sum_{e:e(t+1)=i} \widehat{p}_{t+1}(e) \propto \sum_{e:e(t+1)=i} \widehat{p}_t(e) \exp(-\eta \widehat{\ell}_t(e)) = \sum_{j=1}^{N} \left(\sum_{\substack{e:e(t)=j \\ e(t+1)=i}} \widehat{p}_t(e) \right) \exp(-\eta \ell_t(j)).$$

[1] Indeed this is true because $\ln\frac{1}{1-\rho} = \ln\left(1 + \frac{\rho}{1-\rho}\right) \le \frac{\rho}{1-\rho}$.

Next, we claim that

$$\sum_{\substack{e:e(t)=j\\ e(t+1)=i}} \widehat{p}_t(e) = \begin{cases} (1-\beta)p_t(j) & \text{if } j = i, \\ \frac{\beta}{N-1}p_t(j) & \text{else.} \end{cases} \tag{2}$$

To see this, notice that

$$\sum_{i=1}^{N} \sum_{\substack{e:e(t)=j\\ e(t+1)=i}} \widehat{p}_t(e) = \sum_{e:e(t)=j} \widehat{p}_t(e) = p_t(j)$$

and also for any $i \neq j$,

$$\frac{\sum\limits_{\substack{e:e(t)=j\\ e(t+1)=j}} \widehat{p}_t(e)}{\sum\limits_{\substack{e:e(t)=j\\ e(t+1)=i}} \widehat{p}_t(e)} = \frac{\sum\limits_{\substack{e:e(t)=j\\ e(t+1)=j}} \pi(e(1)) \left(\prod\limits_{\tau=2}^{t} \pi(e(\tau)|e(\tau-1))\right) \pi(j|j) \exp(-\eta \sum\limits_{\tau<t} \ell_\tau(e(\tau)))}{\sum\limits_{\substack{e:e(t)=j\\ e(t+1)=i}} \pi(e(1)) \left(\prod\limits_{\tau=2}^{t} \pi(e(\tau)|e(\tau-1))\right) \pi(i|j) \exp(-\eta \sum\limits_{\tau<t} \ell_\tau(e(\tau)))} = \frac{1-\beta}{\frac{\beta}{N-1}}$$

which together implies Eq. (2). We therefore continue with

$$\begin{aligned} p_{t+1}(i) &\propto \sum_{j=1}^{N} \left(\sum_{\substack{e:e(t)=j\\ e(t+1)=i}} \widehat{p}_t(e) \right) \exp(-\eta \ell_t(j)) \\ &= (1-\beta)p_t(i)\exp(-\eta\ell_t(i)) + \frac{\beta}{N-1} \sum_{j \neq i} p_t(j)\exp(-\eta\ell_t(j)) \\ &= (1-\alpha)p_t(i)\exp(-\eta\ell_t(i)) + \frac{\alpha}{N} \sum_{j=1}^{N} p_t(j)\exp(-\eta\ell_t(j)) \qquad \text{(define } \alpha = \tfrac{N\beta}{N-1}) \end{aligned}$$

which implies

$$p_{t+1}(i) = (1-\alpha)\frac{p_t(i)\exp(-\eta\ell_t(i))}{\sum_{j=1}^{N} p_t(j)\exp(-\eta\ell_t(j))} + \frac{\alpha}{N} \tag{3}$$

With p_1 being the uniform distribution, this provides an efficient and recursive formula for computing p_t! Note that the multiplicative update form of Hedge plays a key role in this derivation. Update rule Eq. (3) is called the fixed-share algorithm [Herbster and Warmuth, 1998]. Despite the somewhat complicated derivation, the final algorithm is in fact extremely simple. When $\alpha = 0$, fixed-share simply recovers Hedge. When $\alpha \neq 0$, fixed-share is mixing some amount of uniform exploration into Hedge, but in a recursive way. Indeed, fixed-share is different from the following update rule which mixes some uniform exploration into Hedge directly

$$p_t(i) \propto (1-\alpha)\frac{\exp(-\eta\sum_{\tau=1}^{t-1}\ell_\tau(i))}{\sum_{j=1}^{N}\exp(-\eta\sum_{\tau=1}^{t-1}\ell_\tau(j))} + \frac{\alpha}{N}.$$

The difference is in fact crucial – in the above update rule, the losses at different time are still treated equally, while in fixed-share, by expanding the recursive formula one can see that roughly speaking, the most recent loss is weighted by $(1-\alpha)$, the second most recent loss is weighted by $(1-\alpha)^2$, so on and so forth. This is very important for getting switching regret or in general for non-stationary environments since recent data is intuitively more useful than data obtained a long time ago.

As a final remark, one can also prove interval regret and even dynamic regret for fixed-share (details omitted). In all cases, parameter tunning (η and α) is a problem, but there are also approaches to fix it.

References

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