Appendix

A Discretization based algorithm

Definition 19 (r-discretization). *An r-discretization or r-net of a bounded set $S \subset \mathbb{R}^d$ is a finite set of points $\mathcal{D}$ such that the Euclidean distance of any point in S is at most r from some point in $\mathcal{D}$.*

Recall that $\mathcal{C} \subset \mathbb{R}^d$ is contained in a ball of radius R. A standard greedy construction gives an r-discretization of size at most $(3R/r)^d$ [Balcan et al., 2018a]. Given the dispersion parameter β, a natural choice is to use a $T^{-\beta}$-discretization as in Algorithm 1.

Theorem 4. *Let $R^{finite}(T,s,N)$ denote the s-shifted regret for the finite experts problem on N experts, for the algorithm used in step 2 of Algorithm 1. Then Algorithm 1 enjoys s-shifted regret $R^{\mathcal{C}}(T,s)$ which satisfies*

$$R^{\mathcal{C}}(T,s) \leq R^{finite}\left(T, s, \left(3RT^{\beta}\right)^d\right) + (sH+L)O(T^{1-\beta}).$$

Proof of Theorem 4. We show we can round the optimal points in $\mathcal{C}$ to points in the $(T^{-\beta})$-discretization $\mathcal{D}$ with a payoff loss at most $(sH+L)T^{1-\beta}$ in expectation. But in $\mathcal{D}$ we know a way to bound regret by $R^{\text{finite}}(T,s,N)$, where N, the number of points in $\mathcal{D}$, is at most $\left(\frac{3R}{T^{-\beta}}\right)^d = \left(3RT^{\beta}\right)^d$.
Let $t_{0:s}$ denote the expert switching times in the optimal offline payoff, and ρ_i^* be the point picked by the optimal offline algorithm in $[t_{i-1}, t_i - 1]$. Consider a ball of radius $T^{-\beta}$ around ρ_i^*. It must have some point $\hat{\rho}_i^* \in \mathcal{D}$. We then must have that $\{u_t \mid t \in [t_{i-1}, t_i - 1]\}$ has at most $O(T^{-\beta}T) = O(T^{1-\beta})$ discontinuities due to β-dispersion, which implies

$$\sum_{t=t_{i-1}}^{t_i-1} u_t(\hat{\rho}_i^*) \geq \sum_{t=t_{i-1}}^{t_i-1} u_t(\rho_i^*) - O(T^{1-\beta})H - L(t_i - t_{i-1})T^{-\beta}$$

Let $\hat{\rho}_t = \hat{\rho}_i^*$ for each $t_{i-1} \leq t \leq t_i - 1$. Summing over i gives

$$\begin{aligned}\sum_{t=1}^{T} u_t(\hat{\rho}_t) &\geq OPT - O(T^{1-\beta})sH - LT^{1-\beta} \\ &= OPT - (sH+L)O(T^{1-\beta})\end{aligned}$$

Now payoff of this algorithm is bounded above by the payoff of the optimal sequence of experts with s shifts

$$\sum_{t=1}^{T} u_t(\hat{\rho}_t) \leq OPT^{finite}$$

Let the finite experts algorithm with shifted regret bounded by $R^{\text{finite}}(T,s,N)$ choose ρ_t at round t. Then, using the above inequalities,

$$\begin{aligned}\sum_{t=1}^{T} u_t(\rho_t) &\geq OPT^{finite} - R^{\text{finite}}(T,s,N) \\ &\geq OPT - (sH+L)O(T^{1-\beta}) - R^{\text{finite}}(T,s,N)\end{aligned}$$

We use this to bound the regret for the continuous case

$$\begin{aligned}R^{\mathcal{C}}(T,s) &= OPT - \sum_{t=1}^{T} u_t(\rho_t) \\ &\leq R^{\text{finite}}(T,s,N) + (sH+L)O(T^{1-\beta})\end{aligned}$$

□

B Counterexamples

We will construct problem instances where some suboptimal algorithms mentioned in the paper suffer high regret.

We first show that the Exponential Forecaster algorithm of [Balcan et al., 2018a] suffers linear s-shifted regret even for $s = 2$. This happens because pure exponential updates may accumulate high weights on well-performing experts and may take a while to adjust weights when these experts suddenly start performing poorly.

Lemma 20. *There exists an instance where Exponential Forecaster algorithm of [Balcan et al., 2018a] suffers linear s-shifted regret.*

Proof. Let $\mathcal{C} = [0,1]$. Define utility functions

$$u^{(0)}(\rho) = \begin{cases} 1 & \text{if } \rho < \frac{1}{2} \\ 0 & \text{if } \rho \geq \frac{1}{2} \end{cases} \quad \text{and } u^{(1)}(\rho) = \begin{cases} 0 & \text{if } \rho < \frac{1}{2} \\ 1 & \text{if } \rho \geq \frac{1}{2} \end{cases}$$

Now consider the instance where $u^{(0)}(\rho)$ is presented for the first $T/2$ rounds and $u^{(1)}(\rho)$ is presented for the remaining rounds. In the second half, with probability at least $\frac{1}{2}$, the Exponential Forecaster algorithm will select a point from $[0, \frac{1}{2}]$ and accumulate a regret of 1. Thus the expected 2-shifted regret of the algorithm is at least $\frac{T}{2} \cdot \frac{1}{2} = \Omega(T)$. Notice that the construction does not depend on the *step size parameter* λ. □

We further look at the performance of Random Restarts EF (Algorithm 4), an easy-to-implement algorithm which looks deceptively similar to Algorithm 2, against this adversary. Turns out Random Restarts EF may not restart frequently enough for the optimal value of the exploration parameter, and have sufficiently long chains of pure exponential updates in expectation to suffer high regret.

Theorem 21. *There exists an instance where Random Restarts EF (Algorithm 4) with parameters λ and α as in Theorem 6 suffers linear s-shifted regret.*

Proof. The probability of pure exponential updates from $t = T/4$ through $t = 3T/4$ is at least

$$(1-\alpha)^{T/2} = \left(1 - \frac{1}{T-1}\right)^{T/2} > \frac{1}{2}$$

for $T > 5$. By Lemma 20, this implies at least $\frac{T}{8}$ regret in this case, and so the expected regret of the algorithm is at least $\frac{T}{16} = \Omega(T)$. □

C Analysis of algorithms

In this section we will provide detailed proofs of lemmas and theorems from Section 4. We will restate them for easy reference.

Lemma 10. ***(Algorithm 2)*** *For each $t \in [T]$, $w_t(\rho) = \mathbb{E}[\hat{w}_t(\rho)]$ and $W_t = \mathbb{E}[\hat{W}_t]$, where the expectations are over random restarts $\mathbf{z}_t = \{z_1, \ldots, z_{t-1}\}$.*

Proof of Lemma 10. $w_t(\rho) = \mathbb{E}[\hat{w}_t(\rho)]$ implies $W_t = \mathbb{E}[\hat{W}_t]$ by Fubini's theorem (recall $\mathcal{C}$ is closed and bounded). $w_t(\rho) = \mathbb{E}[\hat{w}_t(\rho)]$ follows by simple induction on t. In the base case, $\mathbf{z}_1$ is the empty set and $w_1(\rho) = 1 = \hat{w}_t(\rho) = \mathbb{E}[\hat{w}_t(\rho)]$. For $t > 1$,

$$\begin{aligned}
\mathbb{E}[\hat{w}_t(\rho)] =& (1-\alpha)\mathbb{E}[e^{\lambda u_t(\rho)}\hat{w}_{t-1}(\rho)] + \\
& \frac{\alpha}{\textsc{Vol}(\mathcal{C})}\mathbb{E}\left[\int_{\mathcal{C}} e^{\lambda u_t(\rho)}\hat{w}_{t-1}(\rho)d\rho\right] \\
& \qquad \text{(definition of } \hat{w}_t\text{)} \\
=& (1-\alpha)e^{\lambda u_t(\rho)}\mathbb{E}[\hat{w}_{t-1}(\rho)] + \\
& \frac{\alpha}{\textsc{Vol}(\mathcal{C})}\int_{\mathcal{C}} e^{\lambda u_t(\rho)}\mathbb{E}[\hat{w}_{t-1}(\rho)]d\rho \\
& \qquad \text{(expectation is over } \mathbf{z}_t\text{)} \\
=& (1-\alpha)e^{\lambda u_t(\rho)}w_{t-1}(\rho) \\
& + \frac{\alpha}{\textsc{Vol}(\mathcal{C})}\int_{\mathcal{C}} e^{\lambda u_t(\rho)}w_{t-1}(\rho)d\rho \\
& \qquad \text{(inductive hypothesis)} \\
=& w_t(\rho) \\
& \qquad \text{(definition of } w_t\text{)}
\end{aligned}$$

□

Lemma 11. ***(Algorithm 2)*** *W_{T+1} equals the sum*

$$\sum_{s\in[T]} \sum_{t_0=1<t_1\cdots<t_s=T+1} \frac{\alpha^{s-1}(1-\alpha)^{T-s}}{\textsc{Vol}(\mathcal{C})^{s-1}} \prod_{i=1}^{s} \tilde{W}(t_{i-1}, t_i)$$

Proof. We use Lemma 10 to note that $W_{T+1} = \mathbb{E}_{\mathbf{z}_t}[\hat{W}_{T+1}] = \mathbb{E}_{s,\mathbf{t}_s}[\hat{W}_{T+1} \mid s, \mathbf{t}_s]$, where $\hat{W}_{T+1} \mid s, \mathbf{t}_s$ is the total weight of Algorithm 4 at time $T+1$ given restarts occur exactly at $\mathbf{t}_s$, and is deterministic since all weights $\hat{w}_{T+1}(\rho)$ are fixed given exact restart times. We will now show by an induction on s,

$$\hat{w}_{T+1}(\rho) \mid s, \mathbf{t}_s = \tilde{w}(\rho; t_{s-1}, t_s) \prod_{i=1}^{s-1} \frac{\tilde{W}(t_{i-1}, t_i)}{\textsc{Vol}(\mathcal{C})} \quad (3)$$

In other words, we wish to show that $\hat{w}_{T+1}(\rho) \mid s, \mathbf{t}_s$ (weights of Algorithm 4 at time $T+1$ given restarts occur exactly at $\mathbf{t}_s$) can be expressed as the product of weight $\tilde{w}(\rho; t_{s-1}, t_s)$ at ρ of regular Exponential Forecaster since the last restart times the normalized total weights accumulated over previous runs.

For $s = 1$, we have no restarts and

$$\begin{aligned}
\tilde{w}(\rho; t_{s-1}, t_s) \prod_{i=1}^{s-1} \frac{\tilde{W}(t_{i-1}, t_i)}{\textsc{Vol}(\mathcal{C})} &= \tilde{w}(\rho; t_0, t_1) \prod_{i=1}^{0} \frac{\tilde{W}(t_{i-1}, t_i)}{\textsc{Vol}(\mathcal{C})} \\
&= \tilde{w}(\rho; 1, T+1) \\
&= \hat{w}_{T+1}(\rho) \mid 1, \mathbf{t}_1
\end{aligned}$$

For $s > 1$, the last restart occurs at $t_{s-1} > 1$. By inductive hypothesis for time $t_{s-1} - 1$ until which we've had $s - 2$ restarts,

$$\begin{aligned}
\hat{w}_{t_{s-1}-1}(\rho) \mid s, \mathbf{t}_s &= \hat{w}_{t_{s-1}-1}(\rho) \mid s-1, \mathbf{t}_{s-1} \\
&= \tilde{w}(\rho; t_{s-2}, t_{s-1} - 1) \prod_{i=1}^{s-2} \frac{\tilde{W}(t_{i-1}, t_i)}{\textsc{Vol}(\mathcal{C})}
\end{aligned}$$

Due to restart at t_{s-1},

$$\begin{aligned}
\hat{w}_{t_{s-1}}(\rho) \mid s, \mathbf{t}_s &= \frac{\int_{\mathcal{C}} e^{\lambda u_t(\rho)}\hat{w}_{t_{s-1}-1}(\rho)d\rho}{\textsc{Vol}(\mathcal{C})} \\
&= \prod_{i=1}^{s-1} \frac{\tilde{W}(t_{i-1}, t_i)}{\textsc{Vol}(\mathcal{C})}
\end{aligned}$$

It's regular exponential updates from this point to t_s, which gives (3).

Integrating (3) to get $\hat{W}_{T+1} \mid s, \mathbf{t}_s$, and noting probability of $s-1$ restarts at $\mathbf{t}_s$ is $\alpha^{s-1}(1-\alpha)^{T-s}$ completes the proof.

□

Theorem 6. *The s-shifted regret of Algorithm 2 with $\alpha = s/T$ and $\lambda = \sqrt{s(d\log(RT^\beta) + \log(T/s))/T}/H$ is $O(H\sqrt{sT(d\log(RT^\beta) + \log(T/s))} + (sH + L)T^{1-\beta})$.*

Full proof of Theorem 6. We first provide an upper and lower bound to $\frac{W_{T+1}}{W_1}$.

Upper bound: The proof is similar to the upper bound for exponential weighted forecaster in [Balcan et al., 2018a] and uses Lemma 8 for W_t.

$$\begin{aligned}\frac{W_{t+1}}{W_t} &= \frac{\int_{\mathcal{C}} e^{\lambda u_t(\rho)} w_t(\rho) d\rho}{W_t} \\ &= \int_{\mathcal{C}} e^{\lambda u_t(\rho)} \frac{w_t(\rho)}{W_t} d\rho \\ &= \int_{\mathcal{C}} e^{\lambda u_t(\rho)} p_t(\rho) d\rho\end{aligned}$$

Finally use inequalities $e^{\lambda z} \leq 1 + (e^{\lambda} - 1)z$ for $z \in [0, 1]$ and $1 + z \leq e^z$ to get

$$\begin{aligned}\frac{W_{t+1}}{W_t} &\leq \int_{\mathcal{C}} p_t(\rho) \left(1 + (e^{H\lambda} - 1)\frac{u_t(\rho)}{H}\right) d\rho \\ &= 1 + (e^{H\lambda} - 1)\frac{P_t}{H} \\ &\leq \exp\left((e^{H\lambda} - 1)\frac{P_t}{H}\right)\end{aligned}$$

where P_t denotes the expected payoff of the algorithm in round t. Let $P(\mathcal{A})$ be the expected total payoff. Then we can write $\frac{W_{T+1}}{W_1}$ as a telescoping product which gives

$$\begin{aligned}\frac{W_{T+1}}{W_1} = \prod_{t=1}^{T} \frac{W_{t+1}}{W_t} &\leq \exp\left((e^{H\lambda} - 1)\frac{\sum_t P_t}{H}\right) \\ &= \exp\left(\frac{P(\mathcal{A})(e^{H\lambda} - 1)}{H}\right)\end{aligned} \tag{4}$$

Lower bound: Again the proof is similar to [Balcan et al., 2018a] and the major difference is use of Lemma 11.

We first lower bound payoffs of points close to the optimal sequence of experts using dispersion. If the optimal sequence with s shifts has shifts at t_i^* $(1 \leq i \leq s-1)$, by β-dispersion for any $\rho_i \in \mathcal{B}(\rho_i^*, w)$

$$\sum_{t=t_{i-1}^*}^{t_i^*-1} u_t(\rho_i) \geq \sum_{t=t_{i-1}^*}^{t_i^*-1} u_t(\rho_i^*) - kH - L(t_i^* - t_{i-1}^*)w \tag{5}$$

where $w = T^{-\beta}$ and $k = O(T^{1-\beta})$. Summing both sides over $i \in [s-1]$ helps us relate the lower bound to the payoff OPT of the optimal sequence.

$$\begin{aligned}\sum_{i=1}^{s} \sum_{t=t_{i-1}^*}^{t_i^*-1} u_t(\rho_i) &\geq \sum_{i=1}^{s} \sum_{t=t_{i-1}^*}^{t_i^*-1} u_t(\rho_i^*) - kH - L(t_i^* - t_{i-1}^*)w \\ &= OPT - ksH - LTw\end{aligned} \tag{6}$$

Now to lower bound $\frac{W_{T+1}}{W_1}$, we first lower bound W_{T+1}. We use Lemma 11 and lower bound by picking the term corresponding to times of expert shifts in the optimal sequence with s-shifted expert.

$$\begin{aligned}&W_{T+1} \\ &= \sum_{s \in [T]} \sum_{t_0 = 1 < t_1 \cdots < t_s = T+1} \frac{\alpha^{s-1}(1-\alpha)^{T-s}}{\textsc{Vol}(\mathcal{C})^{s-1}} \prod_{i=1}^{s} \tilde{W}(t_{i-1}, t_i) \\ &\geq \frac{\alpha^{s-1}(1-\alpha)^{T-s}}{\textsc{Vol}(\mathcal{C})^{s-1}} \prod_{i=1}^{s} \tilde{W}(t_{i-1}^*, t_i^*)\end{aligned} \tag{7}$$

The product of $\tilde{W}$'s can in turn be lower bounded by restricting attention to points close (i.e. within a ball of radius w centered at optimal expert ρ_i^*) to the optimal sequence. The payoffs of such points was lower-bounded in (5) and (6) in terms of the optimal payoff.

$$\begin{aligned}&\prod_{i=1}^{s} \tilde{W}(t_{i-1}^*, t_i^*) = \prod_{i=1}^{s} \int_{\mathcal{C}} \exp\left(\lambda \sum_{t=t_{i-1}^*}^{t_i^*-1} u_t(\rho)\right) d\rho \\ &\geq \prod_{i=1}^{s} \int_{\mathcal{B}(\rho_i^*, w)} \exp\left(\lambda \sum_{t=t_{i-1}^*}^{t_i^*-1} u_t(\rho)\right) d\rho \\ &\geq \prod_{i=1}^{s} \int_{\mathcal{B}(\rho_i^*, w)} \exp\left(\lambda \sum_{t=t_{i-1}^*}^{t_i^*-1} u_t(\rho_i^*) - kH - L(t_i^* - t_{i-1}^*)w\right) d\rho \\ &= \textsc{Vol}(\mathcal{B}(w))^s \cdot \\ &\quad \exp\left(\sum_{i=1}^{s} \lambda \sum_{t=t_{i-1}^*}^{t_i^*-1} u_t(\rho_i^*) - kH - L(t_i^* - t_{i-1}^*)w\right) \\ &= \textsc{Vol}(\mathcal{B}(w))^s \exp\left(\lambda\big(OPT - ksH - LTw\big)\right)\end{aligned}$$

Plugging into equation (7) we get

$$\begin{aligned}W_{T+1} \geq &\frac{\alpha^{s-1}(1-\alpha)^{T-s}\textsc{Vol}(\mathcal{B}(w))^s}{\textsc{Vol}(\mathcal{C})^{s-1}} \cdot \\ &\exp\left(\lambda\big(OPT - ksH - LTw\big)\right)\end{aligned}$$

Also, $W_1 = \int_{\mathcal{C}} w_1(\rho) d\rho = \textsc{Vol}(\mathcal{C})$. Thus, using the fact that ratio of volume of balls $\mathcal{B}(w)$ and $\mathcal{B}(R)$ in d-dimensions is $(w/R)^d$, and assuming $\mathcal{C}$ is bounded by some ball $\mathcal{B}(R)$.

$$\begin{aligned}\frac{W_{T+1}}{W_1} \geq &\alpha^{s-1}(1-\alpha)^{T-s}\left(\frac{w}{R}\right)^{sd} \cdot \\ &\exp\left(\lambda\big(OPT - ksH - LTw\big)\right)\end{aligned} \tag{8}$$

Putting together: Combining upper and lower bounds from (4) and (8) respectively,

$$\begin{aligned}&\log\left(\alpha^{s-1}(1-\alpha)^{T-s}\right) - sd\log\frac{R}{w} + \\ &\qquad \lambda(OPT - ksH - LTw) \leq \frac{P(\mathcal{A})(e^{H\lambda} - 1)}{H}\end{aligned}$$

which rearranges to

$$OPT - P(\mathcal{A}) \leq P(\mathcal{A}) \frac{(e^{H\lambda} - 1 - H\lambda)}{H\lambda} + \frac{sd \log(R/w)}{\lambda} + ksH + LTw - \frac{\log(\alpha^{s-1}(1-\alpha)^{T-s})}{\lambda}$$

Using $P(\mathcal{A}) \leq HT$ and using $e^z \leq 1 + z + (e-2)z^2$ for $z \in [0, 1]$ we have

$$\begin{aligned} OPT - P(\mathcal{A}) \leq & HT \frac{(e^{H\lambda} - 1 - H\lambda)}{H\lambda} + \frac{sd \log(R/w)}{\lambda} \\ & + ksH + LTw - \frac{\log(\alpha^{s-1}(1-\alpha)^{T-s})}{\lambda} \\ < & H^2 T\lambda + \frac{sd \log(R/w)}{\lambda} + ksH + LTw \\ & - \frac{\log(\alpha^{s-1}(1-\alpha)^{T-s})}{\lambda} \end{aligned}$$

Now we tighten the bound, first w.r.t. α then w.r.t. λ. Note $\min_\alpha - \log(\alpha^{s-1}(1-\alpha)^{T-s})$ occurs for $\alpha_0 = \frac{s-1}{T-1}$ and

$$\begin{aligned} & -\log(\alpha_0^{s-1}(1-\alpha_0)^{T-s}) \\ & \quad = (T-1)\left[-\frac{s-1}{T-1} \log \frac{s-1}{T-1} - \frac{T-s}{T-1} \log \frac{T-s}{T-1} \right] \\ & \quad \leq (s-1) \log e \frac{T-1}{s-1} \end{aligned}$$

(binary entropy function satisfies $h(x) \leq x \ln(e/x)$ for $x \in [0, 1]$). Finally minimizing over λ gives

$$OPT - P(\mathcal{A}) \leq O(H\sqrt{sT(d \log(R/w) + \log(T/s))} + ksH + LTw)$$

for $\lambda = \sqrt{s(d \log(R/w) + \log(T/s))/T}/H$. Plugging back $w = T^{-\beta}$ and $k = O(T^{1-\beta})$ completes the proof. □

The rest of this section is concerned with the analysis of Algorithm 3 for the sparse experts setting.

Lemma 22. *For any $t < T$,*

$$w_T(\rho) \geq \alpha(1-\alpha)^{T-t} \pi_t(\rho) \tilde{w}(\rho; t, T) W_t$$

Proof. Follows using the restart algorithm technique used in Lemmas 11 and 13. Consider the probability of last 'restart' being at time t. Notice this also implies Corollary 14. □

Lemma 23. *Let $\pi_t(\rho) = \sum_{i=1}^t \beta_{i,t} p_i(\rho)$ in Algorithm 3. Then*

$$\pi_t(\rho) = \frac{\alpha_{1,t}}{W_1} + \sum_{i=1}^{t-1} \alpha_{i+1,t} \frac{e^{\lambda u_i(\rho)} w_i(\rho)}{W_{i+1}}$$

where

$$\alpha_{i,t} \geq \frac{1-\alpha}{e_t} \left(e^{-\gamma} + \frac{\alpha}{e_t} \right)^{t-i}$$

and $e_t := \sum_{i=1}^t e^{-\gamma(i-1)}$.

Proof. Notice, by definition of weight update in Algorithm 3,

$$\begin{aligned} p_t(\rho) &= (1-\alpha) \frac{e^{\lambda u_{t-1}(\rho)} w_{t-1}(\rho)}{W_t} + \alpha \sum_{i=1}^{t-1} \beta_{i,t-1} p_i(\rho) \\ &= (1-\alpha) \frac{e^{\lambda u_{t-1}(\rho)} w_{t-1}(\rho)}{W_t} + \alpha \pi_{t-1}(\rho) \end{aligned}$$

This gives us a recursive relation for $\alpha_{i,t}$.

$$\alpha_{i,t} = \begin{cases} \beta_{i,t}(1-\alpha) + \alpha \sum_{j=i+1}^t \beta_{j,t} \alpha_{i,j-1} & \text{if } i > 1 \\ \beta_{i,t} + \alpha \sum_{j=i+1}^t \beta_{j,t} \alpha_{i,j-1} & \text{if } i = 1 \end{cases}$$

Thus for each $1 \leq i \leq t$

$$\alpha_{i,t} \geq \beta_{i,t}(1-\alpha) + \alpha \sum_{j=i+1}^t \beta_{j,t} \alpha_{i,j-1}$$

We proceed by induction on $t - i$. For $i = t$,

$$\alpha_{t,t} \geq \beta_{t,t}(1-\alpha) = \frac{1-\alpha}{e_t} \left(e^{-\gamma} + \frac{\alpha}{e_t} \right)^{t-t}$$

For $i < t$, by inductive hypothesis

$$\begin{aligned} \alpha_{i,t} \geq & \beta_{i,t}(1-\alpha) + \sum_{j=i+1}^t \beta_{j,t} \alpha \alpha_{i,j-1} \\ \geq & (1-\alpha) \frac{e^{-\gamma(t-i)}}{e_t} \\ & + \alpha \frac{1-\alpha}{e_t} \sum_{j=i+1}^t \frac{e^{-\gamma(t-j)}}{e_t} \left(e^{-\gamma} + \frac{\alpha}{e_t} \right)^{j-1-i} \\ = & \frac{1-\alpha}{e_t} \left(e^{-\gamma} + \frac{\alpha}{e_t} \right)^{t-i} \end{aligned}$$

which completes the induction step. □

Corollary 24. *Let $w_t(\rho), W_t$ be as in Algorithm 3 and π_t as in Lemma 13. For each $\tau < \tau' < t$ and any bounded f defined on $\mathcal{C}$.*

$$\begin{aligned} \int_{\mathcal{C}} \pi_t(\rho) f(\rho) d\rho \geq & \frac{\alpha(1-\alpha)^{\tau'-\tau}(1-e^{-\gamma})}{(e^{-\gamma} + \alpha(1-e^{-\gamma}))^{\tau'-t}} \frac{W_\tau}{W_{\tau'}} \cdot \\ & \int_{\mathcal{C}} \pi_\tau(\rho) \tilde{w}(\rho; \tau, \tau') f(\rho) d\rho \end{aligned}$$

Proof. By Lemma 23,

$$\begin{aligned}\int_{\mathcal{C}} \pi_t(\rho) f(\rho) d\rho &= \int_{\mathcal{C}} \pi_t(\rho) f(\rho) d\rho \\ &\geq \int_{\mathcal{C}} \alpha_{\tau',t} \frac{e^{\lambda u_{\tau'-1}(\rho)} w_{\tau'-1}(\rho)}{W_{\tau'}} f(\rho) d\rho \\ &\geq \frac{1-\alpha}{e_t} \left(e^{-\gamma} + \frac{\alpha}{e_t} \right)^{t-\tau'} \frac{1}{W_{\tau'}} \cdot \\ &\quad \int_{\mathcal{C}} e^{\lambda u_{\tau'-1}(\rho)} w_{\tau'-1}(\rho) f(\rho) d\rho \\ &\geq \frac{1-\alpha}{e_t} \left(e^{-\gamma} + \frac{\alpha}{e_t} \right)^{t-\tau'} \frac{\alpha(1-\alpha)^{\tau'-1-\tau} W_\tau}{W_{\tau'}} \cdot \\ &\quad \int_{\mathcal{C}} \pi_\tau(\rho) \tilde{w}(\rho; \tau, \tau') f(\rho) d\rho\end{aligned}$$

where for the last inequality we have used Lemma 22. The lemma then follows by noting

$$\frac{1}{e_t} = \frac{1-e^{-\gamma}}{1-e^{-\gamma t}} \geq 1 - e^{-\gamma}$$

where $e_t = \sum_{i=1}^{t} e^{-\gamma(i-1)}$ as defined in Lemma 23. ☐

Theorem 7. *The (m-sparse, s-shifted) regret of Algorithm 3 is* $O(H\sqrt{T(md\log(RT^\beta) + s\log(mT/s))} + (mH+L)T^{1-\beta})$ *for* $\alpha = s/T$, $\gamma = s/mT$ *and* $\lambda = \sqrt{(md\log(RT^\beta) + s\log(T/s))/T}/H$.

Proof of Theorem 7. Like Theorem 6 we first provide an upper and lower bound to $\frac{W_{T+1}}{W_1}$. The upper bound proof is identical to that of Theorem 6 by replacing Lemma 8 by Lemma 12.

For the lower bound we use Corollaries 14 and 24. Applying corollary 24 repeatedly to collect exponential updates for the times OPT played the same expert lets us use the arguments for Theorem 6 to get Equation **??**. Indeed if $\{(s_i, f_i) \mid 1 \leq i \leq l\}$ are the start and finish times of a particular expert ρ in the OPT sequence, we can use Corollary 14 to write

$$W_{f_l+1} \geq \alpha(1-\alpha)^{f_l+1-s_l} W_{s_l} \tilde{W}(\pi_{s_l}; s_l, f_l+1)$$

Applying Corollary 24 repeatedly now gets us

$$\begin{aligned}W_{f_l+1} \geq & \frac{\alpha^l (1-\alpha)^{\sum_{j=1}^{l} f_j+1-s_j} (1-e^{-\gamma})^{l-1}}{(e^{-\gamma} + \alpha(1-e^{-\gamma}))^{\sum_{j=1}^{l-1} f_j+1-s_{j+1}}} \cdot \\ & \frac{\prod_{i=1}^{l} W_{s_i}}{\prod_{i=1}^{l-1} W_{f_i+1}} \int_{\mathcal{C}} \left(\pi_{s_1}(\rho) \prod_{j=1}^{l} \tilde{w}(\rho; s_j, f_j+1) \right) d\rho\end{aligned}$$

or

$$\begin{aligned}\frac{\prod_{i=1}^{l} W_{f_i+1}}{\prod_{i=1}^{l} W_{s_i}} \geq & \frac{\alpha^l (1-\alpha)^{\sum_{j=1}^{l} f_j+1-s_j} (1-e^{-\gamma})^{l-1}}{(e^{-\gamma} + \alpha(1-e^{-\gamma}))^{\sum_{j=1}^{l-1} f_j+1-s_{j+1}}} \cdot \\ & \int_{\mathcal{C}} \left(\pi_{s_1}(\rho) \prod_{j=1}^{l} \tilde{w}(\rho; s_j, f_j+1) \right) d\rho\end{aligned}$$

Multiplying these inequalities for each of m experts in the optimal sequence gives us $\frac{W_{T+1}}{W_1}$ on the left side. Also note

$$\int_{\mathcal{C}} \pi_t(\rho) f(\rho) d\rho \geq \frac{\alpha_{1,t}}{W_1} \int_{\mathcal{C}} f(\rho) d\rho$$

and, using dispersion as in proof of Theorem 6,

$$\begin{aligned}&\prod_{\text{experts in OPT}} \int_{\mathcal{C}} \left(\prod_{j=1}^{l} \tilde{w}(\rho; s_j, f_j+1) \right) d\rho \geq \\ &\textsc{Vol}(\mathcal{B}(T^{-\beta}))^m \exp\left(\lambda \left(OPT - (mH+L)O(T^{1-\beta})\right)\right)\end{aligned}$$

Putting it all together gives Equation **??**. Combining the lower and upper bounds on $\frac{W_{T+1}}{W_1}$ gives us a bound on $OPT - P(\mathcal{A})$.

$$\begin{aligned}OPT - P(\mathcal{A}) < & H^2 T\lambda + \frac{md\log(RT^\beta)}{\lambda} \\ & + (mH+L)O(T^{1-\beta}) \\ & - \log\left(\frac{\alpha^s (1-\alpha)^T (1-e^{-\gamma})^s}{(e^{-\gamma} + \alpha(1-e^{-\gamma}))^{-mT}} \right) / \lambda\end{aligned}$$

We now chose parameters γ, α, λ to get the tightest regret bound. Note that $-\log(\alpha^s(1-\alpha)^T)$ is minimized for $\alpha = \frac{s}{T+s} = \Theta(\frac{s}{T})$ and $-\log((1-e^{-\gamma})^s(e^{-\gamma} + \alpha(1-e^{-\gamma}))^{mT})$ is minimized for $\gamma = \log\left(\frac{1+s/mT}{1-s\alpha/mT(1-\alpha)}\right) = \Theta(\frac{s}{mT})$. The corresponding minimum values can be bounded as

$$\begin{aligned}-\log(\alpha^s(1-\alpha)^T) &= s\log\frac{T+s}{s} + T\log\left(1+\frac{s}{T}\right) \\ &\leq s\log\frac{T+s}{s} + s \\ &= O\left(s\log\frac{T}{s}\right)\end{aligned}$$

using $\log(1+x) \leq x$, and substituting $e^{-\gamma} = \frac{1-s\alpha/mT(1-\alpha)}{1+s/mT}$

$$\begin{aligned}&-\log((1-e^{-\gamma})^s(e^{-\gamma} + \alpha(1-e^{-\gamma}))^{mT}) \\ &= -s\log\frac{\frac{s}{mT} \cdot \frac{1}{(1-\alpha)}}{1+\frac{s}{mT}} - mT\log\frac{1}{1+\frac{s}{mT}} \\ &\leq s\log\left((1-\alpha)\left(\frac{mT}{s}+1\right)\right) + 1 \\ &= O\left(s\log\frac{mT}{s}\right)\end{aligned}$$

Finally we minimize w.r.t. λ, to obtain the desired regret bound. ☐

D Adaptive Regret

It is known that the fixed share algorithm obtains good adaptive regret for finite experts and OCO

[Adamskiy et al., 2012]. We show that it is the case here as well.

Definition 25. *The τ-adaptive regret (due to [Hazan and Seshadhri, 2007]) is given by*

$$\mathbb{E}\left[\max_{\substack{\rho^*\in\mathcal{C},\\ 1\le r<s\le T, s-r\le\tau}} \sum_{t=r}^{s}(u_t(\rho^*)-u_t(\rho_t))\right]$$

The goal here is to ensure small regret on all intervals of size up to τ simultaneously. Adaptive regret measures how well the algorithm approximates the best expert locally, and it is therefore somewhere between the static regret (measured on all outcomes) and the shifted regret, where the algorithm is compared to a good sequence of experts.

Theorem 26. *Algorithm 2 enjoys $O(H\sqrt{\tau(d\log(R/w)+\log\tau)}+(H+L)\tau^{1-\beta})$ τ-adaptive regret for $\lambda=\sqrt{(d\log(R\tau^{\beta})+\log(\tau))/\tau}/H$ and $\alpha=1/\tau$.*

Proof sketch of Theorem 26. Apply arguments of Theorem 6 to upper and lower bound W_{s+1}/W_r for any interval $[r,s]\subseteq[1,T]$ of size τ. We get

$$\frac{W_{s+1}}{W_r}\le\exp\left(\frac{P(\mathcal{A})(e^{H\lambda}-1)}{H}\right)$$

where $P(\mathcal{A})$ is the expected payoff of the algorithm in $[r,s]$, Also, by Corollary 14 (equivalent for Algorithm 2)

$$\begin{aligned} W_{s+1} &\ge \frac{\alpha(1-\alpha)^{s+1-r}}{\textsc{Vol}(\mathcal{C})}\tilde{W}(r,s)W_r \\ &= \frac{\alpha(1-\alpha)^{\tau}}{\textsc{Vol}(\mathcal{C})}\tilde{W}(r,s)W_r \end{aligned}$$

By dispersion, as in the proof of Theorem 6,

$$\tilde{W}(r,s)\ge\textsc{Vol}(\mathcal{B}(\tau^{-\beta}))\exp\left(\lambda\left(OPT-(H+L)O(\tau^{1-\beta})\right)\right)$$

Putting the upper and lower bounds together gives us a bound on $OPT-P(\mathcal{A})$, which gives the desired regret bound for $\alpha=\frac{1}{\tau}$. □

E Efficient Sampling

In Section 5 we introduced Algorithm 5 for efficient implementation of Algorithm 2 in $\mathbb{R}^d$. We present proofs of the results in that section, and an exact algorithm for the case $d=1$.

Lemma 15. *In Algorithm 2, for $t\ge1$,*

$$W_{t+1}=(1-\alpha)^{t-1}\tilde{W}(1,t+1)+\frac{\alpha}{\textsc{Vol}(\mathcal{C})}\sum_{i=2}^{t}\left[(1-\alpha)^{t-i}W_i\tilde{W}(i,t+1)\right]$$

Algorithm 6 Fixed Share Exponential Forecaster - exact algorithm for one dimension

Input: $\lambda\in(0,1/H]$

1. $W_1=\textsc{Vol}(\mathcal{C})$
2. For each $t=1,2,\dots,T$:

   Estimate $C_{t,j}$ using Lemma 16 for each $1\le j\le t$ using memoized values for weights.

   Sample i with probability $C_{t,i}$.

   Sample ρ with probability proportional to $\tilde{w}(\rho;i,t)$.

   Estimate W_{t+1} using Lemma 15.

Proof of Lemma 15. For $t=1$, first term is $\tilde{W}(1,2)=\int_{\mathcal{C}}e^{\lambda u_1(\rho)}d\rho=W_2$ and second term is zero. Also, by Lemma 8, for $t>1$

$$\begin{aligned} W_{t+1}&=\int_{\mathcal{C}}e^{\lambda u_t(\rho)}w_t(\rho)d\rho\\ &=\int_{\mathcal{C}}e^{\lambda u_t(\rho)}\Bigg[(1-\alpha)e^{\lambda u_{t-1}(\rho)}w_{t-1}(\rho)\\ &\quad+\frac{\alpha}{\textsc{Vol}(\mathcal{C})}\int_{\mathcal{C}}e^{\lambda u_{t-1}(\rho)}w_{t-1}(\rho)d\rho\Bigg]d\rho\\ &=(1-\alpha)\int_{\mathcal{C}}e^{\lambda(u_t(\rho)+u_{t-1}(\rho))}w_{t-1}(\rho)d\rho\\ &\quad+\frac{\alpha}{\textsc{Vol}(\mathcal{C})}W_t\int_{\mathcal{C}}e^{\lambda u_t(\rho)}d\rho \end{aligned}$$

Continue substituting $w_j(\rho)=(1-\alpha)e^{\lambda u_j(\rho)}w_{j-1}(\rho)+\frac{\alpha}{\textsc{Vol}(\mathcal{C})}\int_{\mathcal{C}}e^{\lambda u_j(\rho)}w_{j-1}(\rho)d\rho$ in the first summand until $w_1=1$ to get the desired expression. □

Definition 27. *For $\alpha\ge0$ we say $\hat{A}$ is an (α,ζ)-approximation of A if*

$$Pr\left(e^{-\alpha}A\le\hat{A}\le e^{\alpha}A\right)\ge1-\zeta$$

Lemma 28. *If $\hat{A}$ is an (α,ζ)-approximation of A and $\hat{B}$ is a (β,ζ')-approximation of B, such that $A,B,\hat{A},\hat{B}$ are all positive reals*

1. *$\hat{A}\hat{B}$ is an $(\alpha+\beta,\zeta+\zeta')$-approximation of AB*
2. *$p\hat{A}+q\hat{B}$ is a $(\max\{\alpha,\beta\},\zeta+\zeta')$-approximation of $pA+qB$ for $p,q\ge0$*

Proof. The results follow from union bound on failure probabilities. □

Corollary 29. *For one-dimensional case, we can exactly compute $\tilde{W}(i,j)$, $1\le i<j\le t$, hence W_t at each iteration can be computed in $O(t)$ time using Lemma 15. More generally, if we have a (β,ζ) approximation*

for each $\tilde{W}(i,j),\ 1 \le i < j \le t$, then by Lemma 15 we can compute a $(t\beta, t^2\zeta)$-approximation for W_{t+1}.

Proof. Union bound on failure probabilities of all $\tilde{W}(i,j),\ 1 \le i < j \le t$ gives we have a β approximation for each with probability at least $1 - t^2\zeta$. This covers failure for all terms in $W_i, 2 \le i \le t$. Further, by induction, the error for estimates for W_i is at most $(i-1)\beta$. By Lemma 28, the error for W_{t+1} estimates is at most $t\beta$. □

Lemma 16. *In Algorithm 2, for $t \ge 1$, $p_t(\rho) = \sum_{i=1}^t C_{t,i}\frac{\tilde{w}(\rho;i,t)}{\tilde{W}(i,t)}$. The coefficients $C_{t,i}$ are given by*

$$C_{t,i} = \begin{cases} 1 & i = t = 1 \\ \alpha & i = t > 1 \\ (1-\alpha)\frac{W_{t-1}}{W_t}\frac{\tilde{W}(i,t)}{\tilde{W}(i,t-1)}C_{t-1,i} & i < t \end{cases}$$

and $(C_{t,1}, \dots, C_{t,t})$ lies on the probability simplex Δ^{t-1}.

Proof of Lemma 16. At each iteration, p_t is obtained by mixing $e^{u_t}p_{t-1}$ with the uniform distribution, i.e. we rescale distributions that p_{t-1} was a mixture of and add one more. Another way to view it is to consider a distribution over the sequences of exponentially updated or randomly chosen points. The final probability distribution is the mixture of a combinatorial number of distributions but a large number of them have a proportional density. $C_{t,i}$ are simply sums of mixture coefficients. This establishes the intuition for the expression for p_t and that the mixing coefficients should sum to 1, but we still need to convince ourselves that the coefficients can be computed efficiently.
We proceed by induction on t. For $t = 1$ (using definitions for $w_2(\rho)$ and $w_2(\rho)$)

$$p_1(\rho) = \frac{w_1(\rho)}{W_1} = \frac{1}{\textsc{Vol}(\mathcal{C})} = C_{1,1}\frac{\tilde{w}(\rho;1,1)}{\tilde{W}(1,1)}$$

(recall $\tilde{w}(\rho;1,1) := 1$ and $\tilde{W}(1,1) = \int_{\mathcal{C}} \tilde{w}(\rho;1,1)d\rho$). For the inductive step, we first express p_{t+1} in terms of p_t

$$\begin{aligned} p_{t+1}(\rho) &= \frac{w_{t+1}(\rho)}{W_{t+1}} \\ &= (1-\alpha)\frac{e^{\lambda u_t(\rho)}w_t(\rho)}{W_{t+1}} + \frac{\alpha}{\textsc{Vol}(\mathcal{C})} \\ &= (1-\alpha)\frac{W_t}{W_{t+1}}\frac{e^{\lambda u_t(\rho)}w_t(\rho)}{W_t} + \frac{\alpha}{\textsc{Vol}(\mathcal{C})} \\ &= (1-\alpha)\frac{W_t}{W_{t+1}}e^{\lambda u_t(\rho)}p_t(\rho) + \frac{\alpha}{\textsc{Vol}(\mathcal{C})} \end{aligned}$$

The lemma is now straightforward to see with induction hypothesis.

$$\begin{aligned} &p_{t+1}(\rho) \\ &=(1-\alpha)\frac{W_t}{W_{t+1}}e^{\lambda u_t(\rho)}\left[\sum_{i=1}^t C_{t,i}\frac{\tilde{w}(\rho;i,t)}{\tilde{W}(i,t)}\right] + \frac{\alpha}{\textsc{Vol}(\mathcal{C})} \\ &=\sum_{i=1}^t\left[(1-\alpha)\frac{W_t}{W_{t+1}}C_{t,i}\frac{\tilde{w}(\rho;i,t+1)}{\tilde{W}(i,t)}\right] + \frac{\alpha}{\textsc{Vol}(\mathcal{C})} \\ &=\sum_{i=1}^t\left[\left((1-\alpha)\frac{W_t}{W_{t+1}}\frac{\tilde{W}(i,t+1)}{\tilde{W}(i,t)}C_{t,i}\right)\frac{\tilde{w}(\rho;i,t+1)}{\tilde{W}(i,t+1)}\right] \\ &\quad+\frac{C_{t+1,t+1}}{\textsc{Vol}(\mathcal{C})} \\ &=\sum_{i=1}^t C_{t+1,i}\frac{\tilde{w}(\rho;i,t+1)}{\tilde{W}(i,t+1)} + \frac{C_{t+1,t+1}}{\textsc{Vol}(\mathcal{C})} \end{aligned}$$

Finally noting

$$C_{t+1,t+1}\frac{\tilde{w}(\rho;t+1,t+1)}{\tilde{W}(t+1,t+1)} = C_{t+1,t+1}\frac{1}{\int_{\mathcal{C}}(1)d\rho} = \frac{C_{t+1,t+1}}{\textsc{Vol}(\mathcal{C})}$$

completes the proof.

Thus W_t (by Lemma 15) and $C_{t,i}$ can be computed recursively for logconcave utility functions using integration algorithm from [Lovász and Vempala, 2006]. We can compute them efficiently using Dynamic Programming.
Finally it's straightforward to establish that the coefficients for p_t must lie on the probability simplex Δ^{t-1}. All coefficients are positive, which is easily seen from the recursive relation and noting all weights are positive. Also we know

$$p_t(\rho) = \sum_{i=1}^t C_{t,i}\frac{\tilde{w}(\rho;i,t)}{\tilde{W}(i,t)}$$

Since $p_t(\rho)$ is a probability distribution by definition, integrating both sides over $\mathcal{C}$ gives

$$\int_{\mathcal{C}} p_t(\rho)d\rho = \sum_{i=1}^t C_{t,i}\frac{\int_{\mathcal{C}}\tilde{w}(\rho;i,t)d\rho}{\tilde{W}(i,t)} \qquad or,$$

$$1 = \sum_{i=1}^t C_{t,i}$$

□

Corollary 30. *If we have a (β,ζ) approximation for each $\tilde{W}(i,j),\ 1 \le i < j \le t$, then by Corollary 29 and Lemma 16 we can compute $\hat{C}_{t+1,i}$ which are $(2t\beta, t^2\zeta)$-approximation for each $C_{t+1,i}$.*

Proof. For $i = t$, we know $C_{t,i}$ exactly by Lemma 16. For $i < t$,

$$C_{t,i} = (1-\alpha)^{t-i} \frac{W_i}{W_t} \frac{\tilde{W}(i,t)}{\textsc{Vol}(\mathcal{C})} C_{i,i} \tag{9}$$

In Corollary 29, we show how to compute $((i-1)\beta, (i-1)^2\zeta)$-approximation for W_i and $((t-1)\beta, (t-1)^2\zeta)$-approximation for W_t given (β, ζ) approximations for each $\tilde{W}(i,j)$, $1 \le i < j \le t$. A similar argument using Lemma 28 shows with failure probability at most $t^2\zeta$, plugging in the approximations in equation 9 has at most $(t+i)\beta$ error. □

Theorem 17. *If utility functions are piecewise concave and L-Lipschitz, we can approximately sample a point ρ with probability $p_{t+1}(\rho)$ in time $\tilde{O}(Kd^4T^4)$ for approximation parameters $\eta = \zeta = 1/\sqrt{T}$ and $\lambda = \sqrt{s(d\ln(RT^\beta) + \ln(T/s))/T}/H$ and enjoy the same regret bound as the exact algorithm. (K is number of discontinuities in u_t's).*

Proof of Theorem 17. Based on Lemma 16, we can sample a uniformly random number r in $[0,1]$ and then sample a ρ from one of t distributions (selected based on r) that $p_t(\rho)$ is a mixture of with probability proportional to $C_{t,i}$. The sampling from the exponentials can be done in polynomial time for concave utility functions using sampling algorithm of [Bassily et al., 2014]. At each round we sample from exactly one of t distributions in the sum for p_t in Lemma 16. We compute $(\eta/6T, \zeta/2T^2)$ approximations for $\tilde{W}(i,j)$, $1 \le i < j \le T$ in time $O(T^2K.T_\int)$ where $T_\int$ is the time to integrate a logconcave distribution (at most $\tilde{O}(d^4/\epsilon^2)$ from [Lovász and Vempala, 2006]). These give $(\eta/3, \zeta/2)$-approximation for $C_{t,i}$'s by corollary 30. Finally we run Algorithm 2 from [Balcan et al., 2018a] with approximation-confidence parameters $(\eta/3, \zeta/2)$. With probability at least $1-\zeta$, $C_{t,i}$ estimation and ρ sampling according to $\tilde{w}(\rho; i, t)$ succeeds. If $\hat{\mu}$ denotes output distribution of ρ with approximate sampling, and μ denotes the exact distribution per $p_t(\rho)$, then we show $D_\infty(\hat{\mu}, \mu) \le \eta$. Indeed, for any set of outcomes $E \subset \mathcal{C}$

$$\begin{aligned}\hat{\mu}(E) = \Pr(\hat{\rho} \in E) &= \sum_{i=1}^{t} \Pr(\hat{\rho} \in E \mid E_{i,t}) \Pr(E_{i,t}) \\ &= \sum_{i=1}^{t} \hat{\mu}_i(E) \frac{\hat{C}_{t,i}}{\sum_j \hat{C}_{t,j}}\end{aligned}$$

where $E_{i,t}$ denotes the event that $\tilde{w}(\rho; i, t)$ was used for sampling $p_t(\rho)$, and $\hat{\mu}_i$ corresponds to the distribution for approximate sampling of $\tilde{w}(\rho; i, t)$. Noting that we used $\eta/3$ approximation for $\hat{\mu}_i$ and each $\hat{C}_{t,i}$, we have

$$\hat{\mu}(E) \le \sum_{i=1}^{t} e^{\eta/3} \mu_i(E) e^{2\eta/3} \frac{C_{t,i}}{\sum_j C_{t,j}} = e^\eta \mu(E)$$

Similarly, $\hat{\mu}(E) \ge e^{-\eta}\mu(E)$ and hence $D_\infty(\hat{\mu}, \mu) \le \eta$. Finally we can show (cf. Theorem 12 of [Balcan et al., 2018a]) that with probability at least $1-\zeta$ the expected utility per round of the approximate sampler is at most a $(1-\eta)$ factor smaller than the expected utility per round of the exact sampler. Together with failure probability of ζ, this implies at most $(\eta+\zeta)HT$ additional regret which results in same asymptotic regret as the exact algorithm for $\eta = \zeta = 1/\sqrt{T}$.

To compute the time complexity, we note from [Lovász and Vempala, 2006] that logconcave functions can be integrated in $\tilde{O}(d^4/\epsilon^2)$ and sampled from in $\tilde{O}(d^3)$ time. The time to integrate dominates the complexity, and the overall complexity can be upper bounded by $O(T^2K \cdot d^4/(\eta/T)^2) = O(KT^4d^4)$. Note: The approximate integration and sampling are only needed for multi-dimensional case, for the one-dimensional case we can compute the weights and sample exactly in polynomial time. □

F Lower bounds

We start with a simple lower bound argument for s-shifted regret for prediction with two experts based on a well-known $\Omega(\sqrt{T})$ lower bound argument for static regret. We will then extend it to the continuous setting and use it for the $\Omega(\sqrt{sT})$ part of the lower bound in Theorem 18 in Section 6.

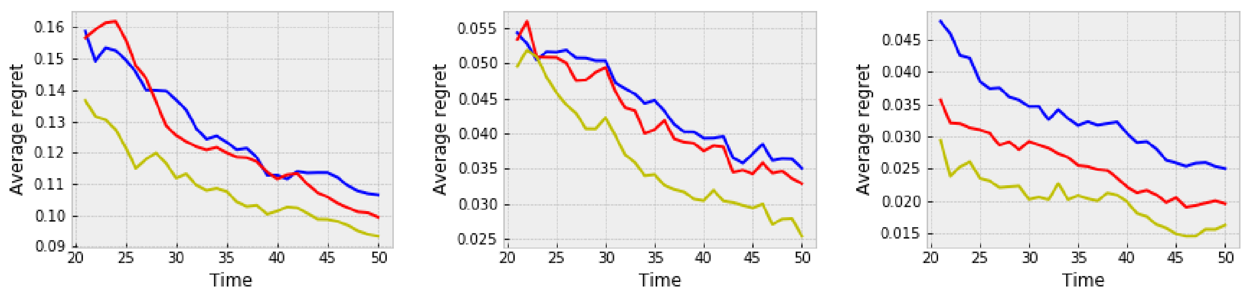
Lemma 31. *For prediction with two experts, there exists a stochastic sequence of losses for which the s-shifted regret of any online learning algorithm satisfies*

$$\mathbb{E}[R_T] \ge \sqrt{sT/8}$$

Proof. Let the two experts predict 0 and 1 respectively at each time $t \in [T]$. The utility at each time t is computed by flipping a coin - with probability 1/2 we have $u(0) = 1, u(1) = 0$ and with probability 1/2 it's $u(0) = 0, u(1) = 1$. Expected payoff for any algorithm $\mathcal{A}$ is

$$P(\mathcal{A}, T) = \mathbb{E}\Big[\sum_{t=1}^{T} u_t(\rho_t)\Big] = \sum_{t=1}^{T} \mathbb{E}[u_t(\rho_t)] = \frac{T}{2}$$

since expected payoff is 1/2 at each t no matter which expert is picked.
To compute shifted regret we need to compare this payoff with the best sequence of experts with $s-1$ switches. We compare with a weaker adversary $\mathcal{A}'$ which is only allowed to switch *up to* $s-1$ times, and



(a) MNIST (b) Omniglot_small_1 (c) Omniglot (full)

Figure 3: Average k-shifted regret vs game duration T for online clustering against k-shifted distributions. Color scheme: **Exponential Forecaster**, **Fixed Share EF**, **Generalized Share EF**

switches at only a subset of fixed times $t_i = iT/s$ to lower bound the regret.

$$\begin{aligned}\mathbb{E}[R_T] &= OPT - P(\mathcal{A}, T) \\ &\geq P(\mathcal{A}', T) - P(\mathcal{A}, T) \\ &= \sum_{t=1}^{T} \mathbb{E}[u_t(\rho'_t)] - \sum_{t=1}^{T} \mathbb{E}[u_t(\rho_t)] \\ &= \sum_{i=0}^{s-1} \sum_{t=t_i+1}^{t_{i+1}} \mathbb{E}[u_t(\rho'_t)] - \mathbb{E}[u_t(\rho_t)]\end{aligned}$$

Now let $P_{i,j} = \sum_{t=t_i+1}^{t_{i+1}} \mathbb{E}[u_t(j)]$ for $i+1 \in [s]$ and $j \in \{0,1\}$

$$\begin{aligned}\sum_{t=t_i+1}^{t_{i+1}} \mathbb{E}[u_t(\rho'_t)] &= \max_{\rho\in\{0,1\}} \sum_{t=t_i+1}^{t_{i+1}} \mathbb{E}[u_t(\rho)] \\ &= \frac{1}{2}\Big[P_{i,0} + P_{i,1} + |P_{i,0} - P_{i,1}|\Big] \\ &= \frac{T}{2s} + |P_{i,0} - T/2s|\end{aligned}$$

using $P_{i,0} + P_{i,1} = T/s$. Thus,

$$\begin{aligned}\mathbb{E}[R_T] &\geq \sum_{i=0}^{s-1} \left[\left(\frac{T}{2s} + |P_{i,0} - T/2s|\right) - \frac{T}{2s}\right] \\ &= \sum_{i=0}^{s-1} |P_{i,0} - T/2s|\end{aligned}$$

Noting $P_{i,0} = \sum_{t=t_i+1}^{t_{i+1}} \mathbb{E}[u_t(0)] = \sum_{t=t_i+1}^{t_{i+1}} \left(\frac{1+\sigma_t}{2}\right)$ where σ_t are Rademacher variables over $\{-1,1\}$ and applying Khintchine's inequality (see for example [Ben-David et al., 2009]) we get

$$\mathbb{E}[R_T] \geq \sum_{i=0}^{s-1} \left| \sum_{t=t_i+1}^{t_{i+1}} \sigma_t/2 \right| \geq \sum_{i=0}^{s-1} \sqrt{T/8s} = \sqrt{sT/8}$$

□

Corollary 32. *We can embed the two-expert setting to get a lower bound for the continuous case.*

Proof. Indeed in Lemma 31 let $\mathcal{C} = [0,1]$, expert 0 correspond to $\rho_1 = 1/4$, expert 1 corresponds to $\rho_2 = 3/4$ and replace the loss functions by

$$u^{(0)}(\rho) = \begin{cases} 1 & \text{if } \rho < \frac{1}{2} \\ 0 & \text{if } \rho \geq \frac{1}{2} \end{cases} \quad \text{and } u^{(1)}(\rho) = \begin{cases} 0 & \text{if } \rho < \frac{1}{2} \\ 1 & \text{if } \rho \geq \frac{1}{2} \end{cases}$$

We can further generalize this while dispersing the discontinuities somewhat. Instead of having all the discontinuties at $\rho = \frac{1}{2}$, we can have discontinuities dispersed say within an interval $[\frac{1}{3}, \frac{2}{3}]$ and still have $\Omega(\sqrt{sT})$ regret. □

G Experiments

We supplement our results in Section 7 by looking at different changing environments and comparing with performance in the static environment setting. We also look at differences between Generalized and Fixed Share EFs.

G.1 Frequently changing environments

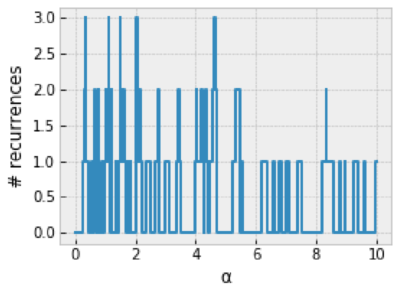
In Section 7 we presented a comparison of our algorithms Fixed Share EF (Algorithm 2) and Generalized Share EF (Algorithm 3) with the Exponential Forecaster algorithm of [Balcan et al., 2018a] for online clustering using well-known datasets. We evaluated the 2-shifted regret for problems where the clustering instance distribution changed exactly once and completely at $T/2$. Here we consider experiments with environments that change more gradually but more frequently.

We consider a sequence of clustering instances drawn from the four datasets. At each time $t \leq T \leq 50$ we sample a subset of the dataset of size 100. For each

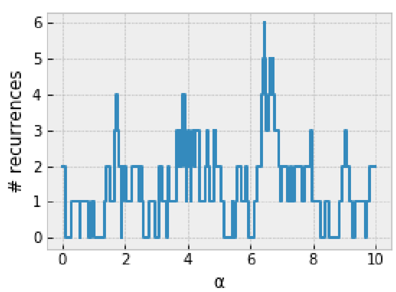
T, we take uniformly random points from all but one classes. The omitted class is changed every T/k rounds, where k is the total number of classes for the dataset. We use parameters $\alpha = \frac{k-1}{T}, \gamma = \frac{1}{T}$ in our algorithms. We determine the hamming cost of $(\bar{\alpha}, 2)$-Lloyds++-clustering for $\alpha \in \mathcal{C} = [0, 10]$ which is used as the piecewise constant loss function.

We compute the average regret against the best offline algorithm with k shifts. In Figure 3 we plot the average of 20 runs for each dataset. The average regret is higher for all algorithms here since the k-shifted baseline is stronger.

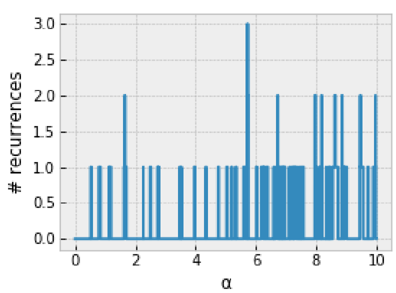
G.2 Generalized vs Fixed Share EFs



(a) MNIST



(b) Omniglot_small_1



(c) Omniglot (full)

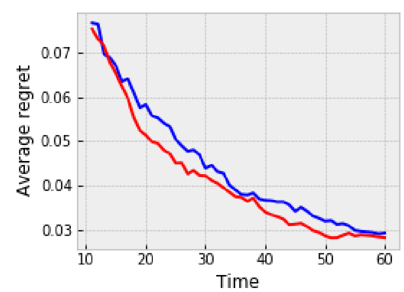
Figure 4: Number of recurrences of various values of α in the top decile across all rounds

We note that Generalized Share EF performs better on most problem instances. This is because it is better able to use recurring patterns in *good* values for the parameter that occur non-contiguously, which depends upon the dataset and the problem instance. We verify this hypothesis by a simple experiment (Figure 4).

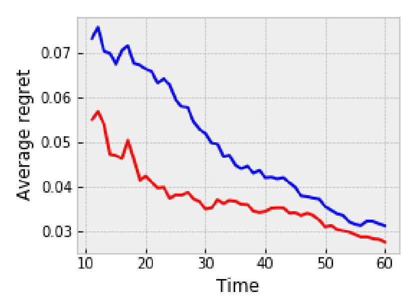
We compute the set of intervals containing the top 10% of the measure of $\alpha \in [0, 10]$ for each t and sum up occurrences of such intervals across all rounds. We observe most recurrences in *Omniglot_small_1* dataset, which explains the large gap between Generalized vs Fixed Share EFs.

G.3 Comparison with static environments

We compare the performance of Fixed Share EF with Exponential Forecaster in static vs dynamic environments on the MNIST dataset. For the changing environment we consider the setting of Section 7, where we present clustering instances for even digits for $t = 1$ through $t = T/2$ and odd digits thereafter. For the static environment we continue to present clustering instances from even labeled digits even after $t = T/2$. We plot the 2-shifted regret in both cases for easier comparison (Figure 5). Note that even though static regret is the more meaningful metric in a static environment, this only changes the baseline and the relative performance of algorithms is unaffected by this choice.

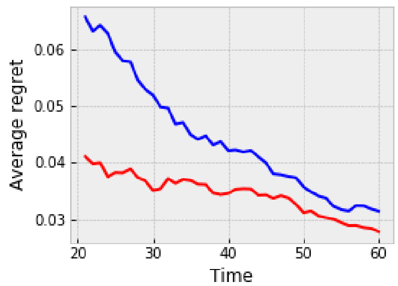


(a) static environment

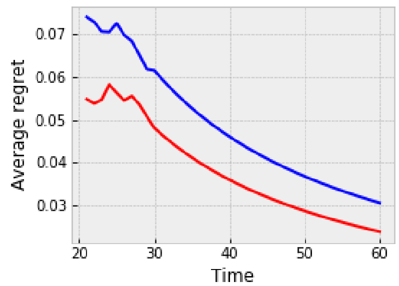


(b) dynamic environment

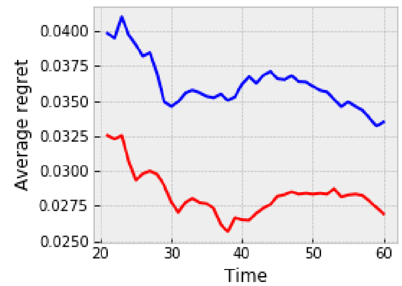
Figure 5: Average 2-shifted regret vs game duration T for online clustering against static/dynamic distributions for the MNIST dataset. Color scheme: **Exponential Forecaster**, **Fixed Share EF**



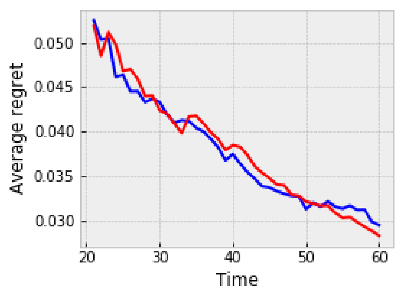
(a) $\{0, 2, 4, 6, 8\}$



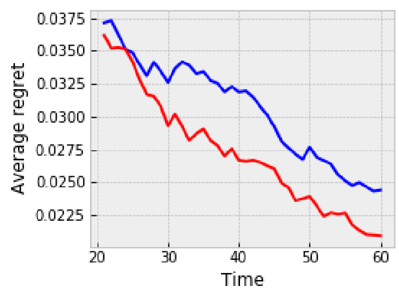
(b) $\{0, 1, 2, 3, 4\}$



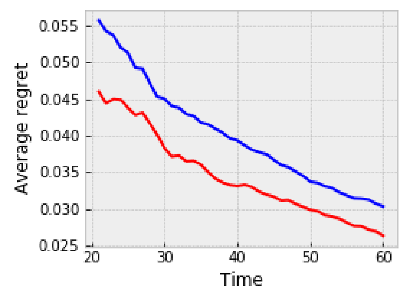
(c) $\{2, 3, 5, 6, 9\}$



(d) $\{1, 3, 4, 8, 9\}$



(e) $\{0, 4, 5, 7, 8\}$



(f) Average

Figure 6: Average 2-shifted regret vs game duration T for online clustering against various dynamic instances for the MNIST dataset. Color scheme: **Exponential Forecaster**, **Fixed Share EF**

Notice that Fixed Share EF is slightly better in the static environment but significantly better in the dynamic environment. It's also worthwhile to note that while the performance of Exponential Forecaster degrades with changing environment, Fixed Share EF actually improves in the dynamic environment since the exploratory updates are more useful.

G.4 Different environments from the same dataset

We look at 2-shifted regret of MNIST clustering instances with the same setting as in Section 7 but with different partitions of clustering classes (i.e. classes used before and after $T/2$). The results are summarized in Figure 6. For each instance we note the set of 5 digits used for drawing uniformly random clustering instances from MNIST till $T/2$, the complement set is used for the remaining rounds. We observe that performance gap between Fixed Share EF and Exponential Forecaster depends not only on the dataset, but also on the clustering instance from the dataset. Across several partitions, Fixed Share EF performs significantly better on average (Figure 6 (f)).