Learning and Games
10716: Advanced Machine Learning
Pradeep Ravikumar

1 Online Learning

In online learning, a learner aims to output a *sequence* of estimates (in contrast to batch learning, where we aim to output a single estimate), by sequentially interacting with nature, which however could potentially be adversarial. In each round t, the learner outputs its estimate $\mathbf{x}_t \in \mathcal{X}$, and the nature/adversary then chooses a loss function $f_t \colon \mathcal{X} \to \mathbb{R}$, and the learner suffers loss $f_t(x_t)$, so that after T rounds, the learner suffers cumulative loss $\sum_{t=1}^{T} f_t(x_t)$.

Obviously the learner wants to suffer the least loss possible, but since nature can choose its loss after seeing the learner's estimate, this might seem like a hopeless dream. And indeed, just minimizing the loss is too hard (without any constraint on how the losses can differ from each other). Accordingly, we ask that the learner choose a sequence of actions $\{\mathbf{x}_t\}_{t=1}^{T}$ such that the following notion of regret is small:

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \inf_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}).$$

This way if nature picks very bad losses f_t, then the baseline to which the learner will be compared to — the best single action chosen in hindsight — could also be bad, and our regret would be small. Assuming the loss functions are bounded, we can say any "learning" is happening only when this regret is $o(T)$, also called sub-linear regret, since the baseline of some random or constant estimate will achieve $O(T)$ regret. But before we analyze this further, let's look at some settings where such online learning naturally crops up.

Examples

Prediction with expert advice: (Freund and Schapire, 97): "A gambler, frustrated by persistent horse-racing losses and envious of his friends' winnings, decides to allow a group of his fellow gamblers to make bets on his behalf. He decides he will wager a fixed sum of money in every race, but that he will apportion his money among his friends based on how well they are doing. Certainly, if he knew psychically ahead of time which of his friends would win the most, he would naturally have that friend handle all his wagers. Lacking such clairvoyance, however, he attempts to allocate each race's wager in such a way that his total winnings for the season will be reasonably close to what he would have won had he bet everything with the luckiest of his friends."

More generally, suppose there are m experts, and in round t, the i-th expert suffers loss $f_t[i]$. The learner's action is a distribution $x_t \in \Delta_m$ over the experts, for which he will suffer loss $f_t(x_t) := \sum_{i=1}^m x_t[i] f_t[i]$. The goal of the learner is to try to do as well as the best expert chosen in hindsight, which is exactly the expression above since the baseline of the best action chosen in hindsight is given as:

$$\min_{x \in \Delta_m} \sum_t \sum_i f_t[i] x[i] = \min_{i \in [m]} \sum_t \sum_i f_t[i],$$

which is exactly the best expert chosen in hindsight.

Portfolio Selection (Hazan, 2023): In each round t, the learner allocates their wealth among m different assets, so that their action is $x_t \in \Delta_m$. The environment, which is the market, then sets the returns $r_t \in \mathbb{R}^m$ of these assets, where $r_t[i]$ is the ratio of asset price in this round vs previous round: if it's greater than one, the asset gained in price, otherwise it fell in price. The ratio of the total wealth of the learner in this round vs previous round is then given by $\sum_i r_t[i] x_t[i]$: the greater this is, the better off the learner. We can thus use as the loss $f_t(x_t) = -\log(\sum_i r_t[i] x_t[i])$. The learner wishes to select portfolio allocations that do as well as the best portfolio they would have chosen in hindsight, that is, *after* observing all the market returns, so that they wish to minimize the regret expression above.

Let's now look at how to achieve sub-linear regret. A key distinction as we will see is whether the domain $\mathcal{X}$ is bounded or not.

1.1 Compact, Convex domain $\mathcal{X}$

Let us first consider the setting where $\mathcal{X}$ is both compact, and convex.

1.1.1 Strongly Convex f_t.

Within this setting, the simplest setting is when the adversary loss functions $\{f_t\}$ are strongly convex.

Myopic. A natural learning strategy is the myopically greedy one, where $\mathbf{x}_t$ is predicted to optimize the previous loss:

$$\mathbf{x}_t = \operatorname*{argmin}_{\mathbf{x} \in \mathcal{X}} f_{t-1}(\mathbf{x}).$$

This would be very natural in both of the examples discussed above. Investors commonly allocate to assets that appreciated in price recently; we might prefer an expert who performed well recently. Common sense suggests this does not always work. Indeed, this need not

achieve sub-linear regret even in this simplest setting. Consider the following example: Suppose $\mathcal{X} \subseteq [-10, 10]$, and that nature sets the following sequence of loss functions

$$f_t(x) = \begin{cases} (x-1)^2, & \text{if } t \text{ is even} \\ (x+1)^2, & \text{if } t \text{ is odd} \end{cases}.$$

Then, a myopic learning strategy would choose the following actions:

$$x_t = \begin{cases} -1, & \text{if } t \text{ is even} \\ 1, & \text{if } t \text{ is odd} \end{cases}.$$

The regret of this algorithm in this case is $\Omega(T)$.

Follow the Leader (FTL). Another natural greedy strategy is the so-called Follow the Leader (FTL), where we choose $\mathbf{x}_t$ based on not just the previous loss, but the average of all losses seen so far:

$$\mathbf{x}_t = \operatorname*{argmin}_{\mathbf{x} \in \mathcal{X}} \sum_{i=1}^{t-1} f_i(\mathbf{x}).$$

This FTL strategy is also sometimes called "fictitious play". This seems a bit more reasonable: we look at a long history to say pick the best performing action. Let κ be the strong convexity parameter of each f_t. Then, it can be shown that the regret is bounded by $O\left(\frac{\log T}{\kappa}\right)$, so that we get sub-linear regret when κ is bounded away from zero.

1.1.2 Convex f_t

We have a simple FTL strategy that achieves near-optimal sublinear regret with strongly convex loss functions. Let us now consider the case where the adversary can select loss functions that are merely convex, and need not be strongly convex. As before, myopic will not achieve sub-linear regret.

Follow the Leader (FTL). But in this setting, even FTL need not have sub-linear regret. Consider the following example. Suppose

$$\mathcal{X} = \{(x_1, x_2) | x_1 + x_2 = 1, x_1 \geq 0, x_2 \geq 0\} \subset \mathbb{R}^2.$$

And suppose the series of loss functions f_t are linear in $\mathbf{x}$ and satisfy $f_t(\mathbf{x}) = \langle g_t, \mathbf{x} \rangle$ for some vector $g_t \in \mathbb{R}^2$. Note that in this setting, $\mathbf{x}_t$ is either $(0, 1)$ or $(1, 0)$. Suppose the adversary chooses g_t as follows

$$g_t = \begin{cases} (1, 0) & \text{if } \mathbf{x}_t = (1, 0) \\ (0, 1) & \text{if } \mathbf{x}_t = (0, 1) \end{cases}$$

The cumulative loss is T, and the loss of the best possible action is:

$$\min_{\mathbf{x}\in\mathcal{X}} \left\langle \sum_{t=1}^{T} g_t, \mathbf{x} \right\rangle = \min \left\{ \sum_{t=1}^{T} g_t[1], \sum_{t=1}^{T} g_t[2] \right\} \leq \frac{T}{2},$$

where the first equality is because the optimum will occur at either $(0,1)$ or $(1,0)$, and the last inequality is because $\sum_{t=1}^{T} g_t[1] + \sum_{t=1}^{T} g_t[2] = T$ by the nature of the adversary loss functions, where each g_t is either $(0,1)$ or $(1,0)$. The regret is thus at least $T/2$. The regret of FTL in this case is thus $\Omega(T)$.

The reason FTL failed here is that our consecutive actions were no longer stable: they could be very far from each other.

Greedy Forecasting. One way in which we might try to fix FTL is a one-step look-forward by simulating what the environment could do in the worst case:

$$\mathbf{x}_t = \arg\min_{\mathbf{x}_t\in\mathcal{X}} \sup_{f_t} \left(\sum_{s=1}^{t-1} f_s(\mathbf{x}_s) + f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{s=1}^{t} f_s(\mathbf{x}) \right).$$

It turns out that even this does not work. Hint: consider loss functions $f(x) = |x - c|$, for $x, c \in [0,1]$.

Follow the Regularized Leader (FTRL). Since we want our iterates to be stable, the natural approach is to regularize the FTL approach, which thus leads to the so-called Follow the Regularized Leader (FTRL) strategy:

$$\mathbf{x}_t = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{X}} \sum_{i=1}^{t-1} f_i(\mathbf{x}) + \eta R(\mathbf{x}),$$

where R is a strongly convex regularizer. Suppose the loss functions are Lipshitz. Let L denote the Lipschitz constant, and D the diameter of the domain $\mathcal{X}$. Then the regret scales as $O(LD\sqrt{T})$. The lower bounds for regret in this setting are $\Omega(LD\sqrt{T})$ [Abernethy et al., 2008]. This shows that the Lipschitz assumption on the loss functions cannot in general be removed.

FTRL-Linearized. Consider the following variant of FTRL where we linearize the previous losses:

$$\mathbf{x}_t = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{X}} \sum_{i=1}^{t-1} \langle \nabla f_i(\mathbf{x}_i), \mathbf{x} \rangle + \eta R(\mathbf{x}),$$

where as before R is a strongly convex regularizer.

Suppose $R(\mathbf{x}) = \|\mathbf{x}\|_2^2$. Then FTRL-linearized can be written as:

$$\mathbf{x}_t = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{X}} \sum_{i=1}^{t-1} \langle \nabla f_i(\mathbf{x}_i), \mathbf{x}\rangle + \eta\|\mathbf{x}\|_2^2.$$

These updates can equivalently be written as

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}(\mathbf{x}_t - \frac{1}{\eta}\nabla f_t(\mathbf{x}_t)).$$

This is also known simply as Online Gradient Descent (OGD).

Let us now consider the case of a general strongly convex regularizer $R(\cdot)$:

$$\mathbf{x}_t = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{X}} \sum_{i=1}^{t-1} \langle \nabla f_i(\mathbf{x}_i), \mathbf{x}\rangle + \eta R(\mathbf{x}),$$

where $R(\mathbf{x})$ is a strongly convex and differentiable regularizer. These updates can be equivalently be written as

$$\nabla R(\mathbf{y}_{t+1}) = \nabla R(\mathbf{y}_t) - \frac{1}{\eta}\nabla f_t(\mathbf{x}_t), \quad \mathbf{x}_{t+1} = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{X}} B_R(\mathbf{x}||\mathbf{y}_{t+1}),$$

where B_R is the Bregman divergence associated with R. This is also more commonly known as Online Mirror Descent (OMD), specifically its *lazy variant*. This is to be contrasted with the so-called *agile variant* of OMD where the updates are given by

$$\nabla R(\mathbf{y}_{t+1}) = \nabla R(\mathbf{x}_t) - \frac{1}{\eta}\nabla f_t(\mathbf{x}_t), \quad \mathbf{x}_{t+1} = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{X}} B_R(\mathbf{x}||\mathbf{y}_{t+1}).$$

Both these algorithms can achieve $O(\sqrt{T})$ regret under appropriate Lipschitz condition on f_t's and boundedness assumption on the domain $\mathcal{X}$ [Hazan, 2016].

Boundedness of $\mathcal{X}$ suffices, even if not compact. In the setting with convex loss functions, compactness is not necessary to achieve sub-linear regret, and it suffices for $\mathcal{X}$ to be bounded. Suppose the domain is open and we are using online projected gradient descent:

$$\mathbf{x}_t = \Pi_{\mathcal{X}}(\mathbf{x}_{t-1} - \eta\nabla f_t(\mathbf{x}_{t-1})) \stackrel{def}{=} \operatorname*{argmin}_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x} - \mathbf{x}_{t-1} + \eta\nabla f_t(\mathbf{x}_{t-1})\|^2.$$

The above problem will not have a minimizer because $\mathcal{X}$ is an open set. But all we need is an approximate minimizer of the above problem to achieve sub-linear regret. If ϵ is the approximation error, then we can achieve $O(LD\sqrt{T} + \epsilon)$ regret [Suggala and Netrapalli, 2019a].

1.2 Compact but Non-convex Domain $\mathcal{X}$, or Non-convex f_t

We now consider the general but more complex setting where either the domain $\mathcal{X}$ or the loss functions $\{f_t\}$ are not convex. Under this setting, no deterministic algorithm can achieve sub-linear regret (i.e., regret which grows slower than T). Consider the following 1D example. Suppose $\mathcal{X} = [-D, D]$. Suppose the adversary chooses loss functions from the following class of 1-Lipshitz functions:

$$\mathcal{F} = \left\{ g_a(\mathbf{x}) := \left[\frac{D}{2} - |\mathbf{x} - a| \right]_+ : a \in [-D, D] \right\},$$

where $[u]_+ = \max\{0, u\}$. Suppose at each round t, once the learner chooses $\mathbf{x}_t$, the adversary picks the loss function $f_t = g_{\mathbf{x}_t}$. It can then be seen that

$$f_t(\mathbf{x}_t) = g_{\mathbf{x}_t}(\mathbf{x}_t) = D/2,$$

so that the loss after T steps is $DT/2$. Whereas for the best action in hindsight, its loss is given as:

$$\min_{\mathbf{x} \in [-D,D]} \sum_{t=1}^{T} f_t(\mathbf{x}) = \min_{\mathbf{x} \in [-D,D]} \sum_{t=1}^{T} \left[\frac{D}{2} - |\mathbf{x} - \mathbf{x}_t| \right]_+ \leq \frac{DT}{4}.$$

To see this: suppose WLOG more than T/2 $\mathbf{x}_t$ are non-negative. For all such $\mathbf{x}_t$, $g_{\mathbf{x}_t}(-D) = 0$. For all the remaining $\mathbf{x}_t$, $g_{\mathbf{x}_t}(-D) \leq D/2$. So $\sum_{i=1}^{T} g_{\mathbf{x}_t}(-D) \leq DT/4$. So the loss of best action is bounded by $DT/4$. The regret is thus bounded by $DT/4 = \Omega(T)$.

Thus, when losses are non-convex, any deterministic strategy can suffer linear regret. A natural strategy in this case is thus to consider a relaxed convex variant of the problem by lifting the domain $\mathcal{X}$ to the convex domain $\mathcal{P}_{\mathcal{X}}$ consisting of distributions over $\mathcal{X}$, and corresponding linear loss functions: $\ell_{f_t}(P) = \mathbb{E}_{\mathbf{x} \sim P}[f_t(\mathbf{x})]$. For this linearized setting, we are thus interested in the regret:

$$\sum_{t=1}^{T} \mathbb{E}_{\mathbf{x} \sim P_t}[f_t(\mathbf{x})] - \inf_{P \in \mathcal{P}_{\mathcal{X}}} \mathbb{E}_{\mathbf{x} \sim P} \left[\sum_i f_i(\mathbf{x}) \right].$$

Note that in this case, the baseline estimate with respect to which we measure regret remains the same since:

$$\inf_{P \in \mathcal{P}_{\mathcal{X}}} \mathbb{E}_{\mathbf{x} \sim P} \left[\sum_i f_i(\mathbf{x}) \right] = \inf_{\mathbf{x} \in \mathcal{X}} \sum_i f_i(\mathbf{x}),$$

due to the linearity of the objective on the LHS. We can thus rewrite the objective as:

$$\sum_{t=1}^{T} \mathbb{E}_{\mathbf{x} \sim P_t}[f_t(\mathbf{x})] - \inf_{\mathbf{x} \in \mathcal{X}} \sum_i f_i(\mathbf{x}).$$

An alternative interpretation of the linearized setting above is that the learner plays *randomized* actions $\mathbf{x}_t \sim P_t$, so that the above linearized regret could then be viewed as the *expected regret* of a sequence of randomized actions by the learner. Note that we have merely linearized the problem, so that while it is convex, it is not strongly so. It is thus natural to resort to an FTRL strategy for sub-linear regret; except that the learner would be choosing distributions rather than individual points in $\mathcal{X}$.

Let $\langle P, f \rangle \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{x}\sim P}\left[f(\mathbf{x})\right]$. Then FTRL in the space of probability distributions is given by

$$P_t = \operatorname*{argmin}_{P\in\mathcal{P}_\mathcal{X}} \sum_{i=1}^{t-1} \langle P, f_i \rangle + \eta R(P),$$

where R is a strongly convex regularizer. This instantiation of mirror descent on $\mathcal{P}_\mathcal{X}$ can be shown to achieve sub-linear regret. A canonical choice for R is *negative entropy*, for which the updates can be written as

$$\nabla R(P_{t+1}) = \nabla R(P_t) - \frac{1}{\eta} f_t.$$

Letting p_{t+1} be the density function of P_{t+1}, the updates can be further rewritten as

$$p_{t+1}(\mathbf{x}) \propto \exp\left(-\frac{1}{\eta}\sum_{i=1}^{t} f_i(\mathbf{x})\right).$$

Krichene et al. [2015] showed that such mirror descent on $\mathcal{P}_\mathcal{X}$ with the entropic regularizer achieves $O(\sqrt{dT\log T})$ expected regret. Note however that when $\{f_t\}$ are non-convex, computing randomized actions via sampling from the distributions $P_{t+1}(\cdot)$ above might be computationally challenging.

Finite Domain: Multiplicative Weights It is instructive to consider the entropic regularization approach when the action space $\mathcal{X}$ is finite. Suppose WLOG $\mathcal{X} = \{1, \ldots, m\}$. In that case, the updates above can be written as:

$$P_{t+1}(i) \propto P_t(i)\exp\left(-\frac{1}{\eta} f_t(i)\right), \quad i \in [m].$$

This learning algorithm is called *Multiplicative Weights*, and has reappeared and been reinvented across different recastings of the online learning problem, ranging over solving sequential games (e.g. boosting as we will see in a few sections), and optimization (devising fast algorithm for LPs and SDPs) among others.

A slightly simpler technique to solve the linearized problem over the space of distributions is via the Follow the Perturbed Leader (FTPL) algorithm [Agarwal et al., 2018, Suggala and

Netrapalli, 2019b]. In this algorithm, the learner predicts

$$\mathbf{x}_t \in \arg\min_{\mathbf{x}\in\mathcal{X}} \sum_{i=1}^{t-1} f_i(\mathbf{x}) - \langle \sigma, \mathbf{x} \rangle ,$$

where $\sigma \in \mathbb{R}^d$ is a random perturbation such that

$$\{\sigma_j\}_{j=1}^d \overset{i.i.d}{\sim} \text{Exp}(\eta),$$

and $\text{Exp}(\eta)$ is the exponential distribution with parameter η. Recall that X is an exponential random variable with parameter η if $P(X \geq s) = \exp(-\eta s)$. When the domain $\mathcal{X}$ is bounded and loss functions $\{f_t\}_{t=1}^T$ are Lipschitz (not necessarily convex), FTPL achieves $O(\sqrt{d^3 T})$ expected regret, for appropriate choice of η [Suggala and Netrapalli, 2019b]. Note that in this case as well, when $\{f_t\}$ are non-convex, solving for the FTPL objective requires solving a non-convex problem, which is computationally challenging in general (though from a practical standpoint, potentially easier than its FTRL counterpart involving sampling). It can be shown that FTPL can also be cast as FTRL for a regularization function that depends on the noise distribution [Suggala and Netrapalli, 2019b].

1.3 Without any assumptions on $\mathcal{X}$ or the loss functions f_t.

When the domain $\mathcal{X}$ is not bounded, note that none of the results above are useful. In particular, regret bounds of FTRL and FTPL scale with the diameter of the domain, and hence would be vacuous for unbounded domains. But there is a very simple strategy, that is applicable without making any assumptions on the domain whatsoever, but under the provision that f_t *was known* to the learner ahead of round t: an optimal strategy for the learner then is to simply predict

$$\mathbf{x}_t \in \arg\min_{\mathbf{x}\in\mathcal{X}} f_t(\mathbf{x}).$$

It is easy to see that this algorithm, known as Best Response (BR), has 0 regret. This is however an impractical algorithm in the general online learning setup, since f_t is not known to the learner at step t prior to making its decision. It can however be used to solve two-player games, as we will see in the sequel.

2 How can we achieve sub-linear regret?

Consider the general online learning setup, where in each round, the learner plays action $x_t \in \mathcal{X}$, and nature provides a loss $f_t(\cdot)$, and the goal is to minimize the cumulative regret

with respect to the best possible action:

$$\sum_{t=1}^{T} f_t(x_t) - \inf_{x \in \mathcal{X}} \sum_{t=1}^{T} f_t(x).$$

It seems amazing that we are able to achieve sub-linear regret even when nature can specify the losses after observing our actions. Why are we able to achieve this? And how do we approach this methodologically from first principles: the methods above work, but can we derive the methods above from first principles?

2.1 Stability Viewpoint

For simplicity, consider convex losses and online gradient descent. We have that the difference in loss suffered by learner compared to that of any action $\mathbf{x}$ is given by:

$$f_t(\mathbf{x}_t) - f_t(\mathbf{x}) \leq g_t^T (x_t - \mathbf{x}),$$

where we use the shorthand $g_t = \nabla f_t(x_t)$. If we are able to bound the sum of the RHS terms over T rounds sub-linearly in T then we would be done, but how are we supposed to do so when the environment can pick f_t adversarially after looking at our action $\mathbf{x}_t$? This is where the online gradient descent steps come in handy: this allows us to bound the RHS via a difference of how close x_{t+1} and x_t are to $\mathbf{x}$: which in turn we can uniformly bound via the diameter of $\mathcal{X}$. Following (Hazan, 2023); if:

$$x_{t+1} = x_t - \eta_t g_t,$$

then:

$$\begin{aligned} \|x_{t+1} - x\|^2 &= \|x_t - x\|^2 + \eta_t^2 \|g_t\|^2 - 2\eta_t g_t^T (x_t - x) \\ 2\eta_t g_t^T (x_t - x) &\leq \frac{1}{\eta_t} \left(\|x_t - x\|^2 - \|x_{t+1} - x\|^2 \right) + \eta_t G^2, \end{aligned}$$

where $\sup \|\nabla f(x)\| \leq G$. Summing above over T rounds we get:

$$\begin{aligned}
2\sum_{t=1}^{T}(f_t(\mathbf{x}_t) - f_t(\mathbf{x})) &\leq 2\sum_{t=1}^{T} g_t^T(x_t - \mathbf{x}) \\
&\leq \sum_{t=1}^{T} \frac{1}{\eta_t}\left(\|x_t - x\|^2 - \|x_{t+1} - x\|^2\right) + \eta_t G^2 \\
&\leq \sum_{t=1}^{T} \|x_t - x\|^2 \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) + G^2 \sum_{t=1}^{T} \eta_t \\
&\leq D^2 \sum_{t=1}^{T} \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) + G^2 \sum_{t=1}^{T} \eta_t \\
&\leq 3DG\sqrt{T},
\end{aligned}$$

where $D = \sup_{x \neq x'} \|x - x'\|$, and by setting $\eta_t = \frac{D}{G\sqrt{t}}$ and using $\sum_{t=1}^{T} 1/\sqrt{t} \leq 2\sqrt{T}$.

2.2 Potential function viewpoint

Let $r_t(x) = f_t(x_t) - f_t(x)$ be the instantaneous regret wrt action x at time t, and let the cumulative regret be: $R_t(x) = \sum_{s=1}^{t} r_t(x)$. Let $\Phi : \mathbb{R}^{\mathcal{X}} \mapsto \mathbb{R}$ be some potential functional, that quantifies any given cumulative regret $R(\cdot)$, via a real-valued scalar. Suppose the learner, at round t plays the randomized action $P_t \in \mathcal{P}_{\mathcal{X}}$ specified as $P_t(x) \propto \nabla\Phi(R_{t-1})(x)$. We then have that that the instantaneous regret at time t is given as:

$$r_t(x) = \int_{x'} P_t(x') f_t(x') dx - f_t(x),$$

so that:

$$\int_x r_t(x) P_t(x) dx = 0,$$

which in turn entails that:

$$\int_x r_t(x) \nabla\Phi(R_{t-1})(x) dx = 0.$$

Thus, the instantaneous r_t is orthogonal to $\Phi(R_{t-1})$, which is an instantiation of the more general so-called Blackwell condition. Loosely, this indicates that *no matter what* loss function f_t might picked in round t, a randomized strategy based on $P(x) \propto \nabla\Phi(R_{t-1})(x)$ will not increase the potential, so that the actions of the learner will likely to regret that is at a local minimum of Φ. In particular, a goal of these online learning algorithms is to satisfy the Blackwell condition:

$$\int_x r_t(x) \nabla\Phi(R_{t-1})(x) dx \leq 0.$$

The Blackwell condition, together with some other regularity conditions on $\Phi(\cdot)$ entail that $\Phi(R_{t-1} + r_t) < \Phi(R_{t-1})$, which in turn entail a bound on the regret. Consider the broad class of potential functions:

$$\Phi(R) = \psi(\int_x \phi(R(x))dx),$$

where $\phi : \mathbb{R} \to \mathbb{R}$ is nonnegative, increasing, and $\psi : \mathbb{R} \to \mathbb{R}$ is nonnegative, increasing, and concave. Then:

$$\begin{aligned}\psi(\phi(\max_x R(x)) &= \psi(\max_x \phi(R(x)) \\ &\leq \psi(\int_x \phi(R(x))dx) \\ &= \Phi(R).\end{aligned}$$

We thus get:

$$\max_x R(x) \leq \phi^{-1}\psi^{-1}(\Phi(R)),$$

so that if we can bound the potential $\Phi(R)$, we can also bound the max regret.

But the algorithms we have studied so far based on FTRL rather than setting weights proportional to some potential function. But FTRL can be connected to the potential based viewpoint via simple duality based arguments. In FTRL, we select the next action via:

$$\min_{P \in \Delta_{\mathcal{X}}} \sum_x R(x)P(x) + \eta \text{reg}(P).$$

We can then define:

$$\Phi(R) := \min_{P \in \Delta_{\mathcal{X}}} \sum_x R(x)P(x) + \eta \text{reg}(P),$$

By pushing the regularization term to a constraint set, we see that the objective is linear. By Danskin's theorem, we then get that the minimizer of the optimization problem is given by the normalized gradient of $\Phi(R)$ as required.

Example: With the entropy regularization $\text{reg}(P) = \sum_x P(x) \log P(x)$ we get:

$$\Phi(R) = \eta \log \sum_x \exp(R(x)/\eta).$$

It can be seen that FTRL and potential gradients both yield the exponential weights algorithm.

2.3 Mixability Loss

The loss incurred by the learner when playing P_t is: $\sum_{\mathbf{x}} P_t(\mathbf{x}) f_t(\mathbf{x})$. But suppose instead the learner suffers a loss, termed mixability loss:

$$\Phi_\eta(P_t, f_t) = -\frac{1}{\eta} \ln(\sum_{\mathbf{x}} P_t(\mathbf{x}) \exp(-\eta f_t(\mathbf{x}))).$$

When choosing exponential weights P_t, we have:

$$\begin{aligned}\sum_{\mathbf{x}} P_t(\mathbf{x}) \exp(-\eta f_t(\mathbf{x}))) &= \frac{\sum_{\mathbf{x}} \sum_{s \le t} \exp(-\eta f_s(x_s))}{\sum_{\mathbf{x}} \sum_{s < t} \exp(-\eta f_s(x_s))} \\ &= \frac{W_t}{W_{t-1}},\end{aligned}$$

so that

$$\begin{aligned}\sum_{s=1}^{t} \Phi_\eta(P_t, f_t) &= -\frac{1}{\eta} \ln \left(\prod_{s=1}^{t} \frac{W_s}{W_{s-1}} \right) \\ &= -\frac{1}{\eta} \ln \frac{W_t}{W_0} \\ &= -\frac{1}{\eta} \ln \frac{\sum_{\mathbf{x}} \exp(-\eta F_t(\mathbf{x}))}{|\mathcal{X}|} \\ \le \min_{\mathbf{x}} F_t(\mathbf{x}) + \frac{1}{\eta} \ln |\mathcal{X}|,\end{aligned}$$

where $F_t(\mathbf{x})$ is the cumulative loss of action $\mathbf{x}$ through the t-th round. Thus denoting the cumulative Φ loss by M_t, we can thus see that:

$$\min_{\mathbf{x}} F_t(\mathbf{x}) \le M_t \le \min_{\mathbf{x}} F_t(\mathbf{x}) + \frac{1}{\eta} \ln |\mathcal{X}|.$$

Thus, if the learner were to suffer the mixability loss $\Phi_\eta(P_t, f_t)$ instead of the actual loss $(\sum_{\mathbf{x}} P_t(\mathbf{x}) f_t(\mathbf{x}))$, then we can easily guarantee that the cumulative loss suffered would be atmost $\frac{1}{\eta} \ln |\mathcal{X}|$ away from that suffered by the best action. But of course the learner does not suffer the mixability loss per se. From here on, there are two ways to proceed; one to bound the true loss as a multiplicative factor of the mixability loss, and one by an additive factor.

Suppose the losses f_t that the environment chooses are drawn from a set $\mathcal{F}$ that satisfies the following "mixability" condition: for all $f_t \in \mathcal{F}, P_t, \eta$, let $c(\eta)$ be the smallest number $c > 0$ such that:

$$\sum_{\mathbf{x}} P_t(\mathbf{x}) f_t(\mathbf{x}) \le c(\eta) \Phi_\eta(f_t, P_t).$$

By summing these up over T rounds we get:

$$\sum_{t=1}^{T}\sum_{\mathbf{x}} P_t(\mathbf{x})f_t(\mathbf{x}) \leq c(\eta)\min_{\mathbf{x}} F_t(\mathbf{x}) + \frac{c(\eta)}{\eta}\ln|\mathcal{X}|,$$

so that we would get a multiplicative factor guarantee. See Cesa-Bianchi and Lugosi [2006], Section 3.5 for an instantiation of the above analysis for the learning from expert advice setting.

We can also ask for additive approximation guarantees:

$$\sum_{\mathbf{x}} P_t(\mathbf{x})f_t(\mathbf{x}) \leq \Phi_\eta(f_t, P_t) + \delta_t.$$

We then get that:

$$\sum_{t=1}^{T}\sum_{\mathbf{x}} P_t(\mathbf{x})f_t(\mathbf{x}) \leq \min_{\mathbf{x}} F_t(\mathbf{x}) + \frac{1}{\eta}\ln|\mathcal{X}| + \sum_{t=1}^{T}\delta_t.$$

One can show (Cesa-Bianchi and Lugosi, 2006, Lemma A.1) that

$$\delta_t \leq \eta/8,$$

so that we get

$$\sum_{t=1}^{T}\sum_{\mathbf{x}} P_t(\mathbf{x})f_t(\mathbf{x}) \leq \min_{\mathbf{x}} F_t(\mathbf{x}) + \frac{1}{\eta}\ln|\mathcal{X}| + \frac{T\eta}{8},$$

so that setting $\eta = \sqrt{\frac{8\ln|\mathcal{X}|}{T}}$, we get sub-linear regret scaling as $\sqrt{0.5T\ln|\mathcal{X}|}$. But it is possible to obtain better bounds for δ_T for specific losses (i.e. by constraining the environment appropriately). See also (Adahedge) where they set the learning rate adaptively as

$$\eta_t = \frac{\ln|\mathcal{X}|}{\sum_{s\leq t}\delta_s},$$

to obtain a tighter bound both theoretically as well as in practice.

3 Two-player Games

Consider the following game between two players. One so-called "row player" playing actions $\mathbf{x} \in \mathcal{X}$, and the other "column player" playing actions $\mathbf{y} \in \mathcal{Y}$. Suppose that when the two

players play actions $\mathbf{x}, \mathbf{y}$ respectively, the row player incurs a loss of $\ell(\mathbf{x}, \mathbf{y}) \in \mathbb{R}$, while the column player incurs a loss of $-\ell(\mathbf{x}, \mathbf{y})$. The sum of the losses for the two players can be seen to be equal to zero, so that such a game is known a two-player *zero-sum game*. It is common in such settings to refer to gain $\ell(\mathbf{x}, \mathbf{y})$ of the column player, rather than its loss of $-\ell(\mathbf{x}, \mathbf{y})$.

A popular example of such a zero-sum game is Rock-Papers-Scissors, where the loss function is given as:

	Rock	Paper	Scissors
Rock	$\frac{1}{2}$	1	0
Paper	0	$\frac{1}{2}$	1
Scissors	1	0	$\frac{1}{2}$

where the rows correspond to row actions $\mathbf{x}$, the columns correspond to column actions $\mathbf{y}$, and the entries correspond to the loss of the row player $\ell(\mathbf{x}, \mathbf{y})$.

Suppose the row player plays action $\mathbf{x}$. The worst loss the row player would then incur is given as: $\max_{\mathbf{y} \in \mathcal{Y}} \ell(\mathbf{x}, \mathbf{y})$. Here, it is as if the column player could select their optimal action after seeing the row player action. Accordingly, a conservative approach for the row player is to select an action that minimizes this worst case loss:

$$\mathbf{x}_{\text{MINMAX}} = \operatorname*{argmin}_{\mathbf{x}} \max_{\mathbf{y}} \ell(\mathbf{x}, \mathbf{y}). \tag{1}$$

The action $\mathbf{x}_{\text{MINMAX}}$ is called the minimax action, and its corresponding worst case loss is called the minmax value of the game.

Now let us consider the game from the perspective of the column player. Suppose the column player plays action $\mathbf{y}$. The least gain the column player would then attain is given as: $\min_{\mathbf{x} \in \mathcal{X}} \ell(\mathbf{x}, \mathbf{y})$. Accordingly, a conservative approach for the column player is to select an action that maximizes this worst case gain:

$$\mathbf{y}_{\text{MAXMIN}} = \operatorname*{argmax}_{\mathbf{y}} \min_{\mathbf{x}} \ell(\mathbf{x}, \mathbf{y}). \tag{2}$$

The action $\mathbf{y}_{\text{MAXMIN}}$ is called the maximin action, and its corresponding worst case gain is called the maxmin value of the game.

In general, the two quantities — minmax and maxmin values of the game — are **not** equal, but the following relationship always holds:

$$\max_{\mathbf{y} \in \mathcal{Y}} \min_{\mathbf{x} \in \mathcal{X}} \ell(\mathbf{x}, \mathbf{y}) \leq \min_{\mathbf{x} \in \mathcal{X}} \max_{\mathbf{y} \in \mathcal{Y}} \ell(\mathbf{x}, \mathbf{y}). \tag{3}$$

Intuitively, in the LHS, the row player gets to choose their action after seeing the column player action, so could achieve a lower loss, compared to the RHS. Indeed, here is a common

setting: suppose the loss is non-negative, and for any $\mathbf{y} \in \mathcal{Y}$, there exists $\mathbf{x} \in \mathcal{X}$ s.t. $\ell(\mathbf{x}, \mathbf{y}) = 0$. Then, it is clear that the maxmin value of the game is zero, whereas the minmax value of the game could well be non-zero.

A key caveat with computing minmax or maxmin values of zero-sum games is that without any additional structure such as convexity, it is computationally difficult in general.

So it is common in game theory to consider a *linearized game* in the space of probability measures, which is in general better-behaved. To set up some notation, for any probability distributions $P_{\mathbf{x}}$ over $\mathcal{X}$, and $P_{\mathbf{y}}$ over $\mathcal{Y}$, define:

$$\ell(P_{\mathbf{x}}, P_{\mathbf{y}}) = \mathbb{E}_{\mathbf{x} \sim P_{\mathbf{x}}, \mathbf{y} \sim P_{\mathbf{y}}} \ell(\mathbf{x}, \mathbf{y}).$$

The minmax and maxmin values of the linearized game and the original game are related as follows:

$$\begin{aligned}
\max_{\mathbf{y} \in \mathcal{Y}} \min_{\mathbf{x} \in \mathcal{X}} \ell(\mathbf{x}, \mathbf{y}) &\overset{(a)}{=} \max_{\mathbf{y} \in \mathcal{Y}} \min_{P_{\mathbf{x}} \in \mathcal{P}_{\mathcal{X}}} \ell(P_{\mathbf{x}}, \mathbf{y}) \\
&\leq \max_{P_{\mathbf{y}} \in \mathcal{P}_{\mathcal{Y}}} \min_{P_{\mathbf{x}} \in \mathcal{P}_{\mathcal{X}}} \ell(P_{\mathbf{x}}, P_{\mathbf{y}}) \leq \min_{P_{\mathbf{x}} \in \mathcal{P}_{\mathcal{X}}} \max_{P_{\mathbf{y}} \in \mathcal{P}_{\mathcal{Y}}} \ell(P_{\mathbf{x}}, P_{\mathbf{y}}) \\
&\overset{(b)}{=} \min_{P_{\mathbf{x}} \in \mathcal{P}_{\mathcal{X}}} \max_{\mathbf{y} \in \mathcal{Y}} \ell(P_{\mathbf{x}}, \mathbf{y}) \leq \min_{\mathbf{x} \in \mathcal{X}} \max_{\mathbf{y} \in \mathcal{Y}} \ell(\mathbf{x}, \mathbf{y}),
\end{aligned}$$

where (b) holds because for any row player action $P_{\mathbf{x}} \in \mathcal{P}_{\mathcal{X}}$, $\max_{P_{\mathbf{y}} \in \mathcal{P}_{\mathcal{Y}}} \ell(P_{\mathbf{x}}, P_{\mathbf{y}})$ is equal to $\max_{\mathbf{y} \in \mathcal{Y}} \ell(P_{\mathbf{x}}, \mathbf{y})$, and correspondingly for (a). One sufficient condition for the values of the linearized and original games to be same is when $\mathcal{X}, \mathcal{Y}$ are convex and compact, and $\ell(\mathbf{x}, \mathbf{y})$ is convex in $\mathbf{x}$ and concave in $\mathbf{y}$, which can be shown using Jensen's inequality.

3.1 Nash Equilibrium

Directly solving for the minmax or maxmin values of the (linearized) min-max games is in general computationally hard, in large part because: (a) these values need not be equal, which limits the set of possible optimization algorithms, and (b) the optimal solutions need not be stable, which makes it difficult for simple optimization problems. It is thus preferable that the two values are equal, and the solutions be stable, which is formalized by the game-theoretic notion of a *Nash equilibrium* (NE). John Von Neumann, a founder of game theory, had noted that he could not foresee there even being a theory of games without a theorem that equates the maxmin and minmax values of the game.

For the original zero-sum game in Equation (1), a pair $(\mathbf{x}^*, \mathbf{y}^*) \in \mathcal{X} \times \mathcal{Y}$ is called a pure strategy NE, if the following holds

$$\max_{\mathbf{y} \in \mathcal{Y}} \ell(\mathbf{x}^*, \mathbf{y}) = \ell(\mathbf{x}^*, \mathbf{y}^*) = \min_{\mathbf{x} \in \mathcal{X}} \ell(\mathbf{x}, \mathbf{y}^*).$$

Intuitively, this says that there is no incentive for any player to change their strategy while the other player keeps hers unchanged. Note that whenever a pure strategy NE exists, the minmax and maxmin values of the game are equal to each other:

$$\min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} \ell(\mathbf{x},\mathbf{y}) \leq \max_{\mathbf{y}\in\mathcal{Y}} \ell(\mathbf{x}^*,\mathbf{y}) \leq \ell(\mathbf{x}^*,\mathbf{y}^*) \leq \min_{\mathbf{x}\in\mathcal{X}} \ell(\mathbf{x},\mathbf{y}^*) \leq \max_{\mathbf{y}\in\mathcal{Y}} \min_{\mathbf{x}\in\mathcal{X}} \ell(\mathbf{x},\mathbf{y}).$$

Since the RHS is always upper bounded by the LHS from (3), the inequalities above are all equalities.

As we discussed above, the minmax and maxmin values of the original game in Equation (1) are in general not equal to each other, and when that is the case, from the above, a pure strategy NE will then not exist for the original game (1).

Instead what often exists is a mixed strategy NE, which is precisely a pure strategy NE of the linearized game. That is, $(P_{\mathbf{x}}{}^*, P_{\mathbf{y}}{}^*) \in \mathcal{P}_{\mathcal{X}} \times \mathcal{P}_{\mathcal{Y}}$ is called a mixed strategy NE of the zero-sum game (1), if

$$\max_{P_{\mathbf{y}}\in\mathcal{P}_{\mathcal{Y}}} \ell(P_{\mathbf{x}}{}^*, P_{\mathcal{Y}}) = \ell(P_{\mathbf{x}}{}^*, P_{\mathbf{y}}{}^*) = \min_{P_{\mathbf{x}}\in\mathcal{P}_{\mathcal{X}}} \ell(P_{\mathbf{x}}, P_{\mathbf{y}}{}^*).$$

As with the original game, if $(P_{\mathbf{x}}{}^*, P_{\mathbf{y}}{}^*)$ is a pure strategy NE of the linearized game, aka, a mixed strategy NE of the original game, then the minmax and maxmin values of the linearized game are equal to each other, and, moreover $P_{\mathbf{x}}{}^*$ is a minimax action and $P_{\mathbf{y}}{}^*$ is a maximin action of the linearized game.

Two critical facets of the linearized game are: (a) linearized game NE (aka mixed strategy NE of original game) exist under less stringent conditions, and (b) are computationally easier to compute or approximate.

3.2 Existence of NE

From an optimization standpoint, a NE is a saddle-point of the minmax objective. And it is known that saddle-points exist when the domains $\mathcal{X}, \mathcal{Y}$ are compact, and the objective $\ell(\cdot,\cdot)$ is convex-concave. If either of these conditions do not hold, then saddle-points need not exist. As an example where the first condition does not hold, consider the game where $\mathcal{X} = \mathcal{Y} = (-1, 1)$, and $\ell(x, y) = x - y$. The game is convex-concave, but a saddle point does not exist. This is because the domains are non-compact. As an example where the second condition does not hold, consider the following game where $\mathcal{X} = \mathcal{Y} = [-1, 1]$, and $\ell(x, y) = (x - y)^2$. The above game is non-convex non-concave, and does not have a saddle-point.

When $\mathcal{X}, \mathcal{Y}$ are compact, the loss function is Lipshitz in at least one of its arguments, then one can show that the minmax and maxmin values of the linearized game in Equation (**??**) are equal to each other (Suggala et al, 2020). Such results are known as minimax theorems,

and studied at length in game theory [Von Neumann et al., 2007, Yanovskaya, 1974, Wald, 1949]. Most classical minimax theorems rely on fixed point theorems, whereas Cesa-Bianchi and Lugosi [2006], (Suggala et al, 2020) present constructive learning-style proofs to prove the minimax theorem, where they present an algorithm which outputs an approximate NE. Under the additional condition that the loss function is bounded, they additionally show that linearized game has a minimax and maximum action. We present one such result in the sequel.

3.3 Algorithms to Compute NE

3.3.1 Online Learning

A popular and widely used approach for solving min-max games is to rely on online learning algorithms [Hazan, 2016, Cesa-Bianchi and Lugosi, 2006]. In this approach, the row (minimization) player and the column (maximization) player play a repeated game against each other. Both the players rely on online learning algorithms to choose their actions in each round of the game, with the objective of minimizing their respective regret. The following proposition shows that this repeated game play converges to a NE.

Proposition 1 *(Suggala et al, 2020) Consider a repeated game between the minimization and maximization players in the linearized game. Let $(P_{\mathbf{x}t}, P_{\mathbf{y}_t})$ be the actions chosen by the players in iteration t. Suppose the actions are such that the regret of each player satisfies*

$$\sum_{t=1}^{T} \ell(P_{\mathbf{x}t}, P_{\mathbf{y}_t}) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} \ell(\mathbf{x}, P_{\mathbf{y}_t}) \leq \epsilon_1(T),$$
$$\max_{\mathbf{y}\in\mathcal{Y}} \sum_{t=1}^{T} \ell(P_{\mathbf{x}t}, \mathbf{y}) - \sum_{t=1}^{T} \ell(P_{\mathbf{x}t}, P_{\mathbf{y}_t}) \leq \epsilon_2(T).$$

Let $P_{\mathbf{x}\text{AVG}}, P_{\mathbf{y}\text{AVG}}$ denote the mixture distributions $\frac{1}{T}\sum_{i=1}^{T} P_{\mathbf{x}i}$ and $\frac{1}{T}\sum_{i=1}^{T} P_{\mathbf{y}_i}$. Then $(P_{\mathbf{x}\text{AVG}}, P_{\mathbf{y}\text{AVG}})$ is an approximate mixed strategy NE of the original game:

$$\ell(P_{\mathbf{x}\text{AVG}}, P_{\mathbf{y}\text{AVG}}) \leq \min_{\mathbf{x}\in\mathcal{X}} \ell(\mathbf{x}, P_{\mathbf{y}\text{AVG}}) + \frac{\epsilon_1(T) + \epsilon_2(T)}{T},$$
$$\ell(P_{\mathbf{x}\text{AVG}}, P_{\mathbf{y}\text{AVG}}) \geq \max_{\mathbf{y}\in\mathcal{Y}} \ell(P_{\mathbf{x}\text{AVG}}, \mathbf{y}) - \frac{\epsilon_1(T) + \epsilon_2(T)}{T}.$$

Note that the above proposition doesn't specify an algorithm to generate the iterates $(P_{\mathbf{x}t}, P_{\mathbf{y}_t})$. All it shows is that as long as both the players rely on algorithms which guarantee sub-linear regret, the iterates converge to a NE. As discussed earlier, there exist several algorithms such as FTRL, FTPL, Best Response (BR), which guarantee sub-linear regret. It is important to

choose these algorithms appropriately, given the domains $\mathcal{X}, \mathcal{Y}$ as our choices impact the rate of convergence to a NE and also the computational complexity of the resulting algorithm. For the setting of convex-concave games, online gradient descent-ascent techniques were classically studied in the optimization community [Nedić and Ozdaglar, 2009, Nemirovski, 2004], and which could also be seen as sub-linear regret online learning procedures for the convex-concave game setting.

For the setting where one of the domains is unbounded, FTRL, FTPL are no longer feasible strategies for the corresponding player, since they have regret bounds that scale with the size of the domain, and can not guarantee sub-linear regret for unbounded domains. However, unlike the general online learning setup, BR is a feasible strategy for such a player, and in fact, likely the only feasible strategy when the corresponding domain is unbounded, since it has 0 regret, without any assumptions on the domain. Recall, in order to use BR, the player requires the knowledge of the future action of the opponent. This can be made possible in the context of min-max games by letting the player choose her action after the other player reveals her action.

We note that even if we could recover the mixed strategy NE, recovering global optima can be NP-hard [see Theorem 9 of Chen et al., 2017].

3.3.2 Myopic Play/Alternating Optimization

When $\mathcal{X}, \mathcal{Y}$ are compact, and the loss function is convex-concave, are there are simpler algorithms can sub-linear regret online learning algorithms? One such algorithm is myopic play, which as we saw does not achieve sub-linear regret in a general online learning setup. In the context of a zero-sum game, when both players play myopically, it need not in general converge to an NE. As a simple example, consider the following game:

$$\min_{x \in [-a,b]} \max_{y \in [-a,b]} xy,$$

for some $b > a > 0$. Suppose the row player starts the myopic play at $x_0 = -a$, then we get the following iterates from the algorithm: $y_0 = -a$, $x_1 = b$, $y_1 = b$, $x_2 = -a, y_2 = -a \ldots$ Clearly, we can not have last-iterate convergence here: $\lim_{t \to \infty}(x_t, y_t)$ doesn't converge to a NE. Even the average of the iterates $\left(\frac{1}{T}\sum_{t=1}^{T} x_t, \frac{1}{T}\sum_{t=1}^{T} y_t\right)$ does not converge to a NE. Interestingly, if only one of the players plays myopically, while the other uses a sub-linear regret online learning strategy, then it suffices for the plays to converge to an NE [Cesa-Bianchi and Lugosi, 2006]. When the loss function is additionally *strongly convex*, it is possible for dual myopic play to converge to an NE. Indeed, such dual myopic play is often called "alternating optimization", and these have been shown to converge to NE for specific losses [REFs].

3.3.3 FTL

Another simple strategy that does not lead to sub-linear regret in a general online learning setup is FTL. Interestingly, when both players play FTL, it can be shown that they converge to an NE [Brown, 1951, Berger, 2007]. But very little is known about the rate of convergence even in simple linear games of the form $\min_{\mathbf{x}\in\Delta_d} \max_{\mathbf{y}\in\Delta_d} \mathbf{x}^T A\mathbf{y}$, where Δ_d is the probability simplex in $\mathbb{R}^d$ [Abernethy et al., 2019]. In some cases, it is known that the algorithm can be exponentially slow to converge [Daskalakis and Pan, 2014].

4 Sequential Game Play

There are a number of caveats with the standard game-theoretic setup above:

- The rules of the game i.e. the loss function in general is unknown
- The column player may not be truly adversarial, and may actually allow for the row player to incur a much smaller loss than the minimax game value

With respect to the latter, think about a poker tournament with expert and rookie poker players. We expect the expert poker players to win much more against rookie players, then in a tournament with only other expert poker players. As another example, going back to the rock-paper-scissors game, in one episode of Simpsons, Bart always thinks that Rock is the best action to take, while Lisa knows that Bart will always pick Rock. The minimax optimal strategy — useful against a worst-case adversary — is to randomize over all actions, which will incur a loss of 1/2. Whereas Lisa should actually take advantage of the fact that Bart is not a truly adversarial player, and choose Paper, which will incur a loss of zero.

To model such a setting, it is instructive to consider a sequential variant of the one-shot zero-sum game above. As before, suppose the row player, which we will also term the learner, chooses actions in $\mathcal{X}$, and the column player, which we will also term the environment, chooses actions in $\mathcal{Y}$. Suppose there is a fixed loss $\ell : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ capturing the rules of the game that is to be played sequentially (e.g. rock-paper-scissors). On round $t = 1, \ldots, T$:

1. The learner chooses their mixed strategy P_t
2. The environment chooses their mixed strategy Q_t
3. Learner observes $\ell(\mathbf{x}, Q_t)$ for all $\mathbf{x} \in \mathcal{X}$
4. Learner suffers loss $\ell(P_t, Q_t)$

The goal of the learner is to minimize regret with respect to best action *chosen in hindsight* i.e. after having observed the sequence of column actions and losses:

$$\sum_{t=1}^{T} \ell(P_t, Q_t) - \min_P \sum_{t=1}^{T} \ell(P, Q_t).$$

We could use any sublinear regret online learning algorithm to choose the learner mixed strategies in each round. If we were to use the entropic regularization/multiplicative weights algorithm, this would entail solving:

$$P_{t+1}(\mathbf{x}) \propto P_t(\mathbf{x}) \exp\left(-\eta \ell(\mathbf{x}, Q_t)\right).$$

With this entropic FTRL algorithm, it can be shown that:

$$\sum_{t=1}^{T} \ell(P_t, Q_t) \leq \min_P \left[a_\eta \sum_{t=1}^{T} \ell(P, Q_t) + c_\eta KL(P, P_1) \right],$$

where P_1 is the first mixed strategy chosen by the learner before seeing anything from the environment, and

$$a_\eta = \frac{\eta}{1 - \exp(-\eta)}$$
$$c_\eta = \frac{1}{1 - \exp(-\eta)}.$$

Suppose $|\mathcal{X}| = m$, and suppose P_1 is set to the uniform distribution over $\{1, \ldots, m\}$. The above guarantee then reduces to:

$$\sum_{t=1}^{T} \ell(P_t, Q_t) \leq \min_P \left[a_\eta \sum_{t=1}^{T} \ell(P, Q_t) + c_\eta \ln m \right].$$

In all the bounds above, η can be any constant; optimizing the bound wrt η yields the optimal value $\eta = \ln\left(1 + \sqrt{\frac{2 \ln m}{T}}\right)$. Substituting above, we get the guarantee:

$$\sum_{t=1}^{T} \ell(P_t, Q_t) \leq \min_P \left[\sum_{t=1}^{T} \ell(P, Q_t) + \Delta_T \right],$$

where

$$\Delta_T = \sqrt{\frac{2 \ln m}{T}} + \frac{\ln m}{T} = O\left(\sqrt{\frac{\ln m}{T}}\right).$$

4.1 Online Prediction

Let us consider one application of the general sequential game play framework above for the setting of online prediction. Let $\mathcal{X}$ be a finite set of inputs, and let $\mathcal{H}$ be a finite set of hypotheses $h : \mathcal{X} \mapsto \{-1, +1\}$. Let $f^* : \mathcal{X} \mapsto \{-1, +1\}$ denote a target classifier, not necessarily in $\mathcal{H}$ defining the correct labels for each instance. The learner then plays the following sequential game of online prediction:

1. Learner observes an instance $\mathbf{x}_t \in \mathcal{X}$ chosen arbitrarily.
2. Learner makes randomized predictions $\widehat{y}_t \in \{-1, +1\}$
3. Learner observes the correct label $f^*(\mathbf{x}_t)$

The goal of the learner is to minimize regret wrt number of mistakes i.e. with respect to zero-one loss, when compared to best baseline in $\mathcal{H}$:

$$\sum_{t=1}^{T} P[\widehat{y}_t \neq f^*(\mathbf{x}_t)] - \min_{h \in \mathcal{H}} \sum_{t=1}^{T} I(h(\mathbf{x}_t) \neq f^*(\mathbf{x}_t)).$$

When using entropic FTRL/multiplicative weights, the learner would play a mixed strategy P_t over the set of hypotheses in $\mathcal{H}$, where:

$$P_{t+1}(h) \propto P_t(h) \exp(-\eta I(h(\mathbf{x}_t) \neq f^*(\mathbf{x}_t))),$$

so that at round t, it would make a randomized prediction by picking $h \sim P_t$, and predicting $h(\mathbf{x}_t)$. By a direct application of the guarantee above, we have that:

$$\sum_{t=1}^{T} P[\widehat{y}_t \neq f^*(\mathbf{x}_t)] \leq \min_{h \in \mathcal{H}} \sum_{t=1}^{T} I(h(\mathbf{x}_t) \neq f^*(\mathbf{x}_t)) + O\left(\sqrt{\frac{\ln |\mathcal{H}|}{T}}\right),$$

so that as T becomes large, the number of mistakes made by the algorithm will be very close to that of the best hypothesis chosen in hindsight after seeing all of the environment inputs, and correct labels!

4.2 Application to Mind-Reading

There are official rock-paper-scissors tournaments, where players face off against each other over rounds of rock paper scissors. This might seem silly, since the NE is known: just randomize over the three actions! But the problem is that humans don't have access to a very good source of randomness within our heads, and in practice behave very non-randomly

even when they try their hardest to behave randomly. Here is how we could use the sequential game play above to beat your friends in a repeated rock-paper-scissors game. Decide on a small set of hypotheses $\mathcal{H}$ that only use the last two rounds of game play to decide the next action. For instance, one hypothesis could pick the action that would be equal to one from the last two plays. And another could imagine that the next action would be different from those in the last two plays. And so on. You can pick the hypothesis class $\mathcal{H}$ to the size that you can keep track of without pen or paper. Then all we have to do is play a regret optimal strategy such as multiplicative weights over this finite hypothesis class. You will then be guaranteed to be close to the best possible hypothesis from $\mathcal{H}$ *chosen in hindsight*. For even simpler games with just two actions, and with hypotheses classes just over the last two game plays, some have found the strategy above to beat 90% of humans in practice.

5 Boosting

The online prediction game above serves as warm-up for a very important game: boosting. As with the online prediction game, we let $\mathcal{X}$ be a finite set of inputs, typically these are the training inputs from your training data. Suppose we use a labeling function $c(\mathbf{x})$ to denote the labels for the inputs in $\mathcal{X}$. And let $\mathcal{H}$ be a space of "weak" hypotheses $h : \mathcal{X} \mapsto \{-1, +1\}$. By weak all we mean is that these hypotheses individually need not be very good. Let us recall the general boosting setup:

1. The booster constructs a distribution Q_t on $\mathcal{X}$
2. There is a weak learner that picks a hypothesis $h_t \in \mathcal{H}$ with error at most $1/2 - \gamma$:

$$P_{\mathbf{x}\sim Q_t}[h_t(\mathbf{x}) \neq c(\mathbf{x})] \leq 1/2 - \gamma.$$

At the end of T rounds, the booster combines the hypotheses via a weighted ensemble: $\text{sign}(\sum_{t=1}^{T} \alpha_t h_t)$.

How do we relate this to a game as in the previous sections? Similar to the online prediction, consider a zero-sum game, where the learner picks mixed strategies P over $\mathcal{H}$, and the environment picks distributions Q over the inputs $\mathcal{X}$. And suppose we use the zero-one loss $\ell(h, \mathbf{x}) = I[h(\mathbf{x}) \neq c(\mathbf{x})$, with the mixed-strategy extension $\ell(P, Q) = \mathbb{E}_{h\sim P, \mathbf{x}\sim Q} I[h(\mathbf{x}) \neq c(\mathbf{x})]$. Applying the min-max theorem to the zero-sum game we get:

$$\begin{aligned}
\min_P \max_{\mathbf{x}\in\mathcal{X}} \ell(P, x) &= \min_P \max_Q \ell(P, Q) \\
&= \max_Q \min_P \ell(P, Q) \\
&= \max_Q \min_{h\in\mathcal{H}} \ell(P, Q).
\end{aligned}$$

Suppose the hypothesis class satisfies the following "weak learning" condition: for any distribution Q over $\mathcal{X}$, there exists a hypothesis $h \in \mathcal{H}$ with error at most $1/2 - \gamma$. This entails that:

$$\max_{Q} \min_{h \in \mathcal{H}} \ell(h, Q) \leq 1/2 - \gamma.$$

From above, this in turn entails that:

$$\min_{P} \max_{\mathbf{x} \in \mathcal{X}} \ell(P, \mathbf{x}) \leq 1/2 - \gamma < 1/2.$$

Since $\ell(P, \mathbf{x}) = \mathbb{E}_{h \sim P} I[h(\mathbf{x}) \neq c(\mathbf{x})]$, the above can be stated as: for any input $\mathbf{x} \in \mathcal{X}$, the weighted majority of hypotheses in $\mathcal{H}$, weighted wrt P, have the correct label. This in turn entails that the weighted majority vote would have the correct label for all $\mathbf{x} \in \mathcal{X}$:

$$\text{sign}\left(\sum_{h \in \mathcal{H}} P(h) h(\mathbf{x})\right) = c(\mathbf{x}).$$

Such a hypothesis class $\mathcal{H}$ where some weighted ensemble has zero error is said to be boostable.

Thus the min-max theorem is just a restatement of the equivalence between the weak learning condition and boostability of a set of hypotheses. Moreover the margin for this weighted majority is:

$$(1/2 + \gamma) - (1/2 - \gamma) = 2\gamma,$$

so that the weak learning "edge" parameter γ also characterizes the margin on the resulting boosted classifier.

Now that we have the zero-sum game, we could then use online learning strategies to solve for the NE of this game. There is however one caveat: strategies such as FTRL would require that we maintain a distribution P over the set of hypotheses, and keep track of losses suffered by each of the hypotheses. This would be very expensive for a large, typically infinite set of hypotheses! It is also not how boosting algorithms such as Adaboost proceed, which track a distribution over the finite training data instances instead.

Since we have an algorithm that tracks distributions for the row player, and we instead care about the action space of the column player, a natural approach is to consider the **dual game**, with loss:

$$\ell'(\mathbf{x}, h) = 1 - \ell(h, \mathbf{x}) = I[h(\mathbf{x}) = c(\mathbf{x})].$$

The row player now chooses a distribution over training instances. And a minimax strategy for the row player is equivalent to a maximin strategy of the original game. When the row player applies the entropic FTRL/multiplicative weights algorithm, we get: On round $t = 1, \ldots, T$:

1. Compute new distribution over training instances:

$$Q_t \propto Q_{t-1} \exp(-\eta I(h_{t-1}(\mathbf{x}) = c(\mathbf{x}))).$$

2. Obtain the weak classifier h_t that solves for:

$$\min_{h \in \mathcal{H}} \ell(h, Q_t),$$

which corresponds to the "best response" strategy.

We know from earlier that if both the row and column players play sub-linear regret strategies, then $\bar{P} = \frac{1}{T}\sum_{t=1}^{T} \delta_{h_t}$ is an (asympotitic) maximin strategy for the dual game, and hence an (asympotitic) minimax strategy for original game.

There is one caveat: we would need the column player to play a regret optimal strategy, such as best response, which might be difficult. But we don't actually need to do that thanks to the weak learnability, and so long as the row player plays a regret optimal strategy (such as entropic FTRL as above). Here's how that plays out. Since the row player plays entropic FTRL, we have that for any choices of $\{h_t\}$:

$$\frac{1}{T}\sum_{t=1}^{T} \ell'(Q_t, h_t) \leq \min_{\mathbf{x} \in \mathcal{X}} \frac{1}{T}\sum_{t=1}^{T} \ell'(\mathbf{x}, h_t) + \Delta_T.$$

But by the weak learning condition, we have that the choices of $\{h_t\}$ satisfy:

$$\ell'(Q_t, h_t) = P_{\mathbf{x} \sim Q_t}[h_t(\mathbf{x}) = c(\mathbf{x})] \geq 1/2 + \gamma.$$

We thus have that:

$$\min_{\mathbf{x} \in \mathcal{X}} \frac{1}{T}\sum_{t=1}^{T} \ell'(\mathbf{x}, h_t) \geq 1/2 + \gamma - \Delta_T > 1/2,$$

for T large enough so that $\Delta_T < \gamma$. Since $\sum_{t=1}^{T} \ell'(\mathbf{x}, h_t)$ is the number of hypotheses that agree with c on $\mathbf{x}$, the above entails that a simple majority vote: $H(\mathbf{x}) = \text{sign}(\sum_{t=1}^{T} h_t(\mathbf{x}))$ would be equal to $c(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$.

In the algorithm above, the multiplicative weights η are all the same. This variant is also called α-boost or ϵ-boost. The Adaboost algorithm further optimizes the weights to minimize the misclassification error, but wasn't fully understood till seminal work by [Breiman, 1999, Friedman et al., 2000] that connected it to coordinate descent of smoothed objectives.

5.1 Smoothed Objectives & Adaboost

We use the short hand notation $L(P, Q)$ to denote $\mathbb{E}_{h \sim P, (x,y) \sim Q}[\mathbb{I}(h(x) \neq y)]$. Letting $\Psi(P) = \sup_{Q \in \mathcal{P}_D} L(P, Q)$, the above optimization problem can be written as

$$\inf_{P \in \mathcal{P}_{\mathcal{H}}} \Psi(P).$$

Thus, rather than use iterative game play strategies as earlier, we could simply use more standard optimization algorithms to solve the program above. A natural way to minimize $\Psi(P)$ is to rely on coordinate descent. However, since $\Psi(P)$ is not a smooth function, such algorithms are not guaranteed to converge to an optimizer. A better technique to use in such cases is to smooth Ψ and perform coordinate descent on the smoothed function [Nesterov, 2005, Abernethy et al., 2014]. Let $\widetilde{\Psi}(P)$ be a smoothed approximation of $\Psi(P)$. A natural technique for constructing $\widetilde{\Psi}(P)$ relies on strongly convex regularizers

$$\widetilde{\Psi}(P) = \sup_{Q \in \mathcal{P}_D} L(P, Q) - \eta R(Q),$$

where $R(Q)$ is a strongly convex function. Following duality between strong convexity and smoothness, it is easy to show that $\widetilde{\Psi}(P)$ is η^{-1} smooth whenever $R(Q)$ is η strongly convex. Moreover, $\widetilde{\Psi}(P)$ is pointwise close to $\Psi(P)$

$$\sup_{P \in \mathcal{P}_{\mathcal{H}}} |\widetilde{\Psi}(P) - \Psi(P)| \leq \eta \sup_{Q \in \mathcal{P}_D} R(Q).$$

Suppose $R(Q) = \sum_i Q_i \log Q_i$, then

$$\widetilde{\Psi}(P) = \eta \log \sum_{i=1}^{n} e^{\frac{L(P,i)}{\eta}},$$

where $L(P, i) = \mathbb{E}_{h \sim P} [\mathbb{I}(h(x_i) \neq y_i)]$. Now consider solving the following optimization problem

$$\inf_{P \in \mathcal{P}_{\mathcal{H}}} \widetilde{\Psi}(P).$$

AdaBoost can be viewed as performing greedy coordinate descent on the following objective [Breiman, 1999, Friedman et al., 2000] where the minimization is over the set of all signed (i.e. unnormalized) measures on $\mathcal{H}$.

Suppose we use coordinate descent with exact line search to minimize $\widetilde{\Psi}(P)$. Then the updates of the algorithm are given by

$$h_t \in \arg\min_{h \in \mathcal{H}} \left\langle \mathbf{e}_h, \nabla \widetilde{\Psi}\left(P_{t-1}\right) \right\rangle, \quad P_t = P_{t-1} + \eta_t h_t,$$

where $\mathbf{e}_h$ is the standard basis vector corresponding to hypothesis h and η_t is chosen using exact line search

$$\eta_t = \operatorname*{argmin}_{\alpha \in \mathbb{R}} \widetilde{\Psi}(P_{t-1} + \alpha h_t).$$

A simple calculation shows that the above updates can be written as

$$h_t \in \arg\min_{h \in \mathcal{H}} \mathbb{E}_{(x,y) \sim Q_t} [-yh(x)], \quad Q_t(i) \propto e^{-\sum_{s=1}^{t-1} \eta^{-1} \eta_s y_i h_s(x_i)},$$

where $\eta_t = \frac{\eta}{2} \log \frac{\sum_{i:h_t(x_i)=y_i} Q_{t-1}(i)}{\sum_{i:h_t(x_i)\neq y_i} Q_{t-1}(i)}$. This is exactly the AdaBoost algorithm. Note that this is not exactly a coordinate descent algorithm. This is because in greedy coordinate descent one chooses direction h_t as follows

$$h_t \in \arg\max_{h\in\mathcal{H}} \left| \left\langle \mathbf{e}_h, \nabla\widetilde{\Psi}\left(P_{t-1}\right)\right\rangle \right|.$$

The AdaBoost algorithm presented above instead solves $\arg\min_{h\in\mathcal{H}} \left\langle \mathbf{e}_h, \nabla\widetilde{\Psi}\left(P_{t-1}\right)\right\rangle$. However, both the updates are equivalent if we assume that $-h \in \mathcal{H}$ whenever $h \in \mathcal{H}$.

Breiman [1999] further distinguished between two classes of algorithms. Their so-called Type I algorithms consider unnormalized P, and solve for:

$$\min_{P\in\mathbb{R}^{\mathcal{H}}} \sum_{z\in D} \rho(L(P,z)) - C\|P\|_1),$$

for some constant $C > 0$, and some univariate function $\rho : \mathbb{R} \mapsto \mathbb{R}$ s.t. $\rho(t) \to \infty$ as $t \to \infty$.

Examples. Adaboost is a Type I algorithm using $\rho(t) = \exp(t)$ and $C = 1/2$.

Their so-called Type II algorithms minimize the above surrogate objective:

$$G(P) = \sum_{z\in D} \rho(L(P,z)),$$

for some function $\rho : \mathbb{R} \mapsto \mathbb{R}$ where $\rho'(t) \geq 0$, and continuous ρ'' s.t. $\sup_{t\in[0,1]} \rho''(t) \geq 0$. Type II algorithms aim more to minimize $L(P, U_D)$ where U is the uniform distribution over D, rather than the minimax objective $\max_{Q\in\mathcal{P}_D} L(P,Q)$.

Examples. Arc-x4 [Breiman, 1996] is a Type II algorithm with $\rho(t) = t^5$.

References

Jacob Abernethy, Peter L Bartlett, Alexander Rakhlin, and Ambuj Tewari. Optimal strategies and minimax lower bounds for online convex games. *Tech. Report*, 2008.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Arun Sai Suggala and Praneeth Netrapalli. Online non-convex learning: Following the perturbed leader is optimal. *arXiv preprint arXiv:1903.08110*, 2019a.

Walid Krichene, Maximilian Balandat, Claire Tomlin, and Alexandre Bayen. The hedge algorithm on a continuum. In *International Conference on Machine Learning*, pages 824–832, 2015.

Naman Agarwal, Alon Gonen, and Elad Hazan. Learning in non-convex games with an optimization oracle. *arXiv preprint arXiv:1810.07362*, 2018.

Arun Sai Suggala and Praneeth Netrapalli. Online non-convex learning: Following the perturbed leader is optimal. *CoRR*, abs/1903.08110, 2019b. URL `http://arxiv.org/abs/1903.08110`.

Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

John Von Neumann, Oskar Morgenstern, and Harold William Kuhn. *Theory of games and economic behavior (commemorative edition)*. Princeton university press, 2007.

EB Yanovskaya. Infinite zero-sum two-person games. *Journal of Soviet Mathematics*, 2(5): 520–541, 1974.

Abraham Wald. Statistical decision functions. *The Annals of Mathematical Statistics*, pages 165–205, 1949.

Angelia Nedić and Asuman Ozdaglar. Subgradient methods for saddle-point problems. *Journal of optimization theory and applications*, 142(1):205–228, 2009.

Arkadi Nemirovski. Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

Robert S Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. Robust optimization for non-convex objectives. In *Advances in Neural Information Processing Systems*, pages 4705–4714, 2017.

George W Brown. Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, 13(1):374–376, 1951.

Ulrich Berger. Brown's original fictitious play. *Journal of Economic Theory*, 135(1):572–578, 2007.

Jacob D. Abernethy, Kevin A. Lai, and Andre Wibisono. Fictitious play: Convergence, smoothness, and optimism. *ArXiv*, abs/1911.08418, 2019.

Constantinos Daskalakis and Qinxuan Pan. A counter-example to karlin's strong conjecture for fictitious play. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 11–20. IEEE, 2014.

Leo Breiman. Prediction games and arcing algorithms. *Neural computation*, 11(7):1493–1517, 1999.

Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

Jacob Abernethy, Chansoo Lee, Abhinav Sinha, and Ambuj Tewari. Online linear optimization via smoothing. In *Conference on Learning Theory*, pages 807–823, 2014.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.