1 Predictive regret minimization

So far, we have talked extensively about regret minimization algorithms, which we modeled as object with the following interface:

- `NextStrategy()` returns a strategy $x^t \in X \subseteq \mathbb{R}^n$;
- `ObserveLoss(ℓ)` receives a utility vector $\ell^t$ that is meant to evaluate the strategy $x^t$ that was last output.

However, a recent trend in online learning has been concerned with constructing devices that can incorporate predictions of the next utility vector $\ell^t$ in the decision making [Chiang et al., 2012, Rakhlin and Sridharan, 2013a,b]. We call these devices predictive regret minimizers.

We incorporate predictivity by modifying the interface above. In particular, for a predictive regret minimizer we modify `NextStrategy` to now be as in the next definition.

**Definition 1.1 (Predictive regret minimizer).** Let $X$ be a set. A predictive regret minimizer for $X$ is an algorithm that interacts with the environment through two operations:

- `NextStrategy(m)` returns the next strategy $x^t \in X$, given a prediction $m^t \in \mathbb{R}^n$ of the next utility $\ell^t$;
- `ObserveLoss(ℓ)` receives a utility vector $\ell^t$ that is meant to evaluate the strategy $x^t$ that was last output. Specifically, the device incurs a utility equal to $(\ell^t)^\top x^t$. As we mentioned, the utility vector $\ell^t$ can depend on all past strategies that were output by the regret minimizer (even including $x^t$, assuming the latter is deterministic).

Just like for regular (i.e., non-predictive) regret minimizers, the quality metric for a predictive regret minimizer is its cumulative regret, defined as the quantity

$$ R_T := \max_{x \in X} \left\{ \sum_{t=1}^T (\ell^t)^\top x - (\ell^t)^\top x^t \right\}. $$

The goal for a “good” predictive regret minimizer should guarantee a superior regret bound than a non-predictive regret minimizer, especially if $m^t \approx \ell^t$ at all times $t$. Algorithms exist that can guarantee this. For instance, it is always possible to guarantee that

$$ R_T = O \left( 1 + \sum_{t=1}^T ||\ell^t - m^t||^2 \right) $$

*Computer Science Department, Carnegie Mellon University. gfarina@cs.cmu.edu.
when \( \mathcal{X} \) is convex and compact, which implies that the regret stays constant when \( m^t \) is clairvoyant. In fact, even stronger regret bounds can be attained, as we discuss next.

### 1.1 Regret bounded by Variation in Utilities (RVU)

A popular definition for a desirable regret goal for a predictive regret minimizer was given by Syrgkanis et al. [2015]. We restate it here with minor modifications.

**Definition 1.2 (RVU regret bound).** A predictive regret minimizers satisfies the RVU regret bound with parameters \( \alpha, \beta, \gamma \in \mathbb{R}_\geq 0 \) with respect to norm \( \| \cdot \| \) if at all times \( T \),

\[
R_T \leq \alpha + \beta \sum_{t=1}^{T} \| \ell_t^t - m_t^t \|^2 - \gamma \sum_{t=2}^{T} \| x_t^t - x_{t-1}^t \|^2.
\]

(2)

In Section 1.2 we show that predictive regret minimizers that satisfy the RVU regret bound can be used in self play to converge to a bilinear saddle point at the accelerated convergence (saddle point gap) rate of \( O(1/T) \). We will then introduce predictive regret minimizers for generic convex and compact sets in Section 2 and show that they satisfy the RVU regret bound.

### 1.2 Accelerated convergence to saddle points via predictive regret minimization

We have seen in Lecture 3 that the idea behind using regret minimization to converge to a bilinear saddle point

\[
\max_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} x^\top Ay
\]

is to use self play. Back then, we instantiated two regret minimization algorithms, \( \mathcal{R}_X \) and \( \mathcal{R}_Y \), for the domains of the maximization and minimization problem, respectively. At each time \( t \) the two regret minimizers output strategies \( x^t \) and \( y^t \), respectively. Then, they receive as feedback the vectors \( \ell_X^t, \ell_Y^t \) defined as

\[
\ell_X^t := Ay^t, \quad \ell_Y^t := -A^\top x^t,
\]

where \( A \) is Player 1’s payoff matrix (see also Figure 1).

Figure 1: The flow of strategies and utilities in regret minimization for games. The symbol \( \Box \) denotes computation/construction of the utility vector.

We then proved that the average strategies

\[
x^T := \frac{1}{T} \sum_{t=1}^{T} x^t, \quad y^T := \frac{1}{T} \sum_{t=1}^{T} y^t
\]

converge to a saddle point of (3) in the sense that the saddle-point gap of \((x^T, y^T)\) vanishes according to

\[
\gamma(x^T, y^T) \leq \frac{R_X^T + R_Y^T}{T}.
\]
As most non-predictive regret minimizers guarantee $O(\sqrt{T})$ regret in the worst case, the self-play scheme defined above guarantees convergence to Nash equilibrium at the rate $O(1/\sqrt{T})$.

We now show that by using predictive regret minimizers that satisfy the RVU bound (2), we can accelerate convergence from $O(1/\sqrt{T})$ to $O(1/T)$. The key is to use as predictions the previous utility vectors, that is,

$$m^t_X := \ell^t_X, \quad m^t_Y := \ell^t_Y.$$  

Then, the gap $\gamma(\bar{x}^T, \bar{y}^T)$ satisfies

$$\gamma(\bar{x}^T, \bar{y}^T) \leq \frac{R^T_X + R^T_Y}{T},$$

$$\leq \frac{1}{T} \left( \alpha + \beta \sum_{t=1}^T \| Ay^t - Ay^{t-1} \|_2^2 - \gamma \sum_{t=2}^T \| x^t - x^{t-1} \|_2^2 \right)$$

$$+ \frac{1}{T} \left( \alpha + \beta \sum_{t=1}^T \| A^T x^t - A^T x^{t-1} \|_2^2 - \gamma \sum_{t=2}^T \| y^t - y^{t-1} \|_2^2 \right)$$

$$\leq \frac{2\alpha}{T} + \frac{\beta}{T} \frac{\| A \|_2^{2 \beta} - \gamma}{T} \sum_{t=1}^T \| y^t - y^{t-1} \|_2^2 + \frac{\beta}{T} \frac{\| A \|_2^{2 \beta} - \gamma}{T} \sum_{t=1}^T \| x^t - x^{t-1} \|_2^2,$$ (5)

where in the second inequality we plugged in the RVU regret bounds for $R_X$ and $R_Y$ (Definition 1.2) and the third inequality follows by noting that the operator norm $\| \cdot \|_{op}$ of a linear function is equal to the operator norm of its transpose.

Inequality (5) immediately implies that if $\| A \|_2^{2 \beta} \leq \gamma / \beta$ then the saddle point gap $\gamma(\bar{x}^T, \bar{y}^T)$ vanishes at the rate $O(1/T)$. Since rescaling $A$ in (3) does not change the set of saddle points, we can always rescale $A$ before applying the self-play algorithm above to guarantee accelerated $O(1/T)$ convergence.

## 2 Predictive FTRL and predictive OMD

**Follow-the-regularized-leader (FTRL)** [Shalev-Shwartz and Singer, 2007] and **online mirror descent (OMD)** are the two best known oracles for the online linear optimization problem. Their predictive variants are relatively new and can be traced back to the works by Rakhlin and Sridharan [2013a] and Syrgkanis et al. [2015].

Algorithms 1 and 2 give pseudocode for predictive FTRL and predictive OMD. The non-predictive variants of FTRL and OMD algorithms correspond to predictive FTRL and predictive OMD when the prediction $m^t$ is set to the $0$ vector at all $t$. In both algorithm, $\eta > 0$ is an arbitrary step size parameter, $X \subseteq \mathbb{R}^n$ is a convex and compact set, and $\varphi : X \to \mathbb{R}_{\geq 0}$ is a differentiable and 1-strongly convex regularizer (with respect to some norm $\| \cdot \|$). The symbol $D_{\varphi}(\cdot, \cdot)$ used in OMD denotes the Bregman divergence associated with $\varphi$, a standard surrogate notion of distance in convex optimization, defined as

$$D_{\varphi}(x \| c) := \varphi(x) - \varphi(c) - \nabla \varphi(c)^T (x - c) \quad \forall x, c \in X.$$  

**Remark 2.1.** The divergence $D_{\varphi}(x \| c)$ is a generalization of the typical notion of distance between $x$ and $c$. When the regularizer $\varphi$ is set to $\varphi(\cdot) = \frac{1}{2} \| \cdot \|^2_2$ (which is 1-strongly convex with respect to the
Algorithm 1: (Predictive) FTRL

Data: $\mathcal{X} \subseteq \mathbb{R}^n$ convex and compact set
$\varphi: \mathcal{X} \to \mathbb{R}_{\geq 0}$ strongly convex regularizer
$\eta > 0$ step-size parameter

1 $L^0 \leftarrow 0 \in \mathbb{R}^n$

2 function NextStrategy($m^t$)
   | $\vartriangleright$ Set $m^t = 0$ for non-predictive version
   | return arg max$_{\hat{x} \in \mathcal{X}} \{ (L^t - 1 + m^t)^T \hat{x} - \frac{1}{\eta} \varphi(\hat{x}) \}$

3 function ObserveUtility($\ell^t$)
   $L^t \leftarrow L^t - 1 + \ell^t$

Algorithm 2: (Predictive) OMD

Data: $\mathcal{X} \subseteq \mathbb{R}^n$ convex and compact set
$\varphi: \mathcal{X} \to \mathbb{R}_{\geq 0}$ strongly convex regularizer
$\eta > 0$ step-size parameter

1 $z^0 \in \mathcal{X}$ such that $\nabla \varphi(z^0) = 0$

2 function NextStrategy($m^t$)
   $\vartriangleright$ Set $m^t = 0$ for non-predictive version
   return arg max$_{\hat{x} \in \mathcal{X}} \{ m^t^T \hat{x} - \frac{1}{\eta} D\varphi(\hat{x} \parallel z^t - 1) \}$

3 function ObserveUtility($\ell^t$)
   $z^t \leftarrow \arg max_{\hat{z} \in \mathcal{X}} \{ \ell^t \parallel \hat{z} - \frac{1}{2\eta} D\varphi(\hat{z} \parallel z^t - 1) \}$


Euclidean norm), then

\[ D\varphi(x \parallel c) = \varphi(x) - \varphi(c) - \nabla \varphi(c)^T (x - c) \]
\[ = \frac{1}{2} \|x\|_2^2 - \frac{1}{2} \|c\|_2^2 - c^T (x - c) \]
\[ = \frac{1}{2} \|x\|_2^2 + \frac{1}{2} \|c\|_2^2 - c^T x \]
\[ = \frac{1}{2} \|x - c\|_2^2 \]

and so we recover that $D\varphi(x \parallel c)$ is half of the squared Euclidean distance between $x$ and $c$.

Remark 2.2. In light of the above remark, when OMD is set up with $\varphi(\cdot) = \|\cdot\|_2^2$, the computation of $z^{t+1}$ (Line 5 of Algorithm 2) amounts to a step of projected gradient ascent. Indeed,

\[
\arg max_{\hat{z} \in \mathcal{X}} \left\{ \ell^T \hat{z} - \frac{1}{2\eta} D\varphi(\hat{z} \parallel z^t - 1) \right\} = \arg max_{\hat{z} \in \mathcal{X}} \left\{ \ell^T \hat{z} - \frac{1}{2\eta} \|\hat{z} - z^t - 1\|_2^2 \right\} = \arg max_{\hat{z} \in \mathcal{X}} \left\{ (\eta \ell^t)^T \hat{z} - \frac{1}{2} \|\hat{z} - z^t - 1\|_2^2 \right\} = \arg max_{\hat{z} \in \mathcal{X}} \left\{ (z^{t-1} + \eta \ell^t)^T \hat{z} - \frac{1}{2} \|\hat{z}\|_2^2 \right\} = \arg max_{\hat{z} \in \mathcal{X}} \left\{ \frac{1}{2} \|\hat{z} \parallel (z^{t-1} + \eta \ell^t)\|_2^2 \right\} = \text{Proj}_{\mathcal{X}}(z^{t-1} + \eta \ell^t).
\]

Remark 2.3. In the non-predictive version of OMD, $m^t = 0$ at all times $t$. In that case, the proximal
step on Line 3 in Algorithm 2 reduces to
\[
\hat{x}^t = \arg \max_{\hat{x} \in \mathcal{X}} \left\{ -\frac{1}{\eta} D_{\varphi}(\hat{x} \parallel z^{t-1}) \right\} = \arg \min_{\hat{x} \in \mathcal{X}} D_{\varphi}(\hat{x} \parallel z^{t-1}) = z^{t-1}.
\]

2.1 Regret guarantee

Predictive FTRL and predictive OMD satisfy the following regret bound.

**Proposition 2.1.** Let \( \Omega \) denote the range of \( \varphi \) over \( \mathcal{X} \), that is, \( \Omega := \max_{x, x' \in \mathcal{X}} \{ \varphi(x) - \varphi(x') \} \). At all times \( T \), the regret cumulated by predictive FTRL (Algorithm 1) and predictive OMD (Algorithm 2) compared to any strategy \( \hat{x} \in \mathcal{X} \) is bounded as
\[
R_T \leq \frac{\Omega}{\eta} + \eta \sum_{t=1}^{T} \| \ell^t - m^t \|^2 - \frac{1}{c \eta} \sum_{i=2}^{T} \| x^i - x^{i-1} \|^2,
\]
where \( c = 4 \) for FTRL and \( c = 8 \) for OMD, and where \( \| \cdot \|_* \) denotes the dual of the norm \( \| \cdot \| \) with respect to which \( \varphi \) is 1-strongly convex. In other words, predictive FTRL satisfies the RVU regret bound with parameters \( (\Omega/\eta, \eta, 1/4\eta) \) and predictive OMD satisfies the RVU regret bound with parameters \( (\Omega/\eta, \eta, 1/8\eta) \).

**Remark 2.4.** Proposition 2.1 immediately implies that, by appropriately setting the step size parameter (for example, \( \eta = T^{-1/2} \)), predictive FTRL and predictive OMD guarantee \( R_T = O(T^{1/2}) \) at all times \( T \). Hence, predictive FTRL and predictive OMD are regret minimizers.

3 Distance-generating functions for tree-form decision problems

Predictive FTRL and predictive OMD, as described above, can be applied to any convex and compact set \( \mathcal{X} \), provided a 1-strongly convex regularizer \( \varphi \) for \( \mathcal{X} \) has been chosen. A safe choice is always \( \varphi(\cdot) = \frac{1}{2} \| \cdot \|_2^2 \), which is 1-strongly convex with respect to the Euclidean norm \( \| \cdot \|_2 \). In that case, we have seen earlier that (non-predictive) OMD reduces to online gradient ascent. However, the choice of regularizer \( \varphi \) has practical implications, because it can make solving the different optimization subproblems involved in the algorithms (for example, Line 5 in Algorithm 2) easy or hard in practice depending on the choice.

Ideally, the choice of regularizer \( \varphi \) makes the computation of the following two quantities as easy as possible. In particular we define the following.

**Definition 3.1 ("Nice" regularizer).** Let \( \mathcal{X} \subseteq \mathbb{R}^n \) be convex and compact. We say that a 1-strongly convex regularizer \( \varphi : \mathcal{X} \to \mathbb{R} \) is "nice" if the following quantities can both be computed in linear time in the dimension \( n \):

- the gradient \( \nabla \varphi(x) \) of \( d \) at any point \( x \in \mathcal{X} \); and
• the gradient of the convex conjugate $\varphi^*$ of $\varphi$ at any point $g \in \mathbb{R}^n$:

$$\nabla \varphi^*(g) = \arg \max_{x \in \mathcal{X}} \{ g^\top x - \varphi(x) \}.$$ 

When the regularizer is nice, then the optimization subproblem on line Line 3 of predictive FTRL can be computed in linear time by noticing that

$$\arg \max_{\hat{x} \in \mathcal{X}} \left\{ (\eta L^{-1} + \eta m^t)^\top \hat{x} - \varphi(\hat{x}) \right\} = \nabla \varphi^*(\eta L^{-1} + \eta m^t).$$

For Line 3 in predictive OMD, we have

$$\arg \max_{\hat{x} \in \mathcal{X}} \left\{ (m^t)^\top \hat{x} - \frac{1}{\eta} D_{\varphi}(\hat{x} \parallel z^{t-1}) \right\} = \arg \max_{\hat{x} \in \mathcal{X}} \left\{ (\eta m^t)^\top \hat{x} - D_{\varphi}(\hat{x} \parallel z^{t-1}) \right\}$$

$$= \arg \max_{\hat{x} \in \mathcal{X}} \left\{ (\eta m^t)^\top \hat{x} - \varphi(\hat{x}) + \nabla \varphi(z^{t-1})^\top \hat{x} \right\}$$

$$= \arg \max_{\hat{x} \in \mathcal{X}} \left\{ (\eta m^t + \nabla \varphi(z^{t-1}))^\top \hat{x} - \varphi(\hat{x}) \right\}$$

$$= \nabla \varphi^*(\eta m^t + \nabla \varphi(z^{t-1})),$$

which can therefore be evaluated in linear time when $\varphi$ is nice. Similarly, for Line 5 we have, using the same derivation,

$$\arg \max_{\hat{z} \in \mathcal{X}} \left\{ (\ell^t)^\top \hat{z} - \frac{1}{\eta} D_{\varphi}(\hat{z} \parallel z^{t-1}) \right\} = \nabla \varphi^*(\eta \ell^t + \nabla \varphi(z^{t-1})).$$

For many sets of interest in game theory, “nice” regularizer are known. For example, for a simplex domain a very appealing regularizer is the negative entropy distance generating function

$$\Delta^n \ni (x_1, \ldots, x_n) \mapsto \sum_{i=1}^n x_i \log x_i.$$ 

The construction of a “nice” regularizer for sequence-form strategy polytopes is significantly more involved, but a modification/generalization of negative entropy with specific weights can be shown to work [Farina et al., 2021].

**Definition 3.2 (Dilatable global entropy).** Consider a tree-form sequential decision process, with the usual notation. The *dilatable global entropy distance generating function* $\tilde{\varphi}$ is the function $\tilde{\varphi} : Q \to \mathbb{R}_{\geq 0}$ defined as

$$\tilde{\varphi} : Q \ni x \mapsto \sum_{j \in \mathcal{J}} \sum_{a \in A_j} w_{ja} x[ja] \log x[ja],$$

where each coefficient $\gamma_j \geq 1$ ($j \in \mathcal{J}$) is defined recursively as

$$\gamma_j := 1 + \max_{a \in A_j} \left\{ \sum_{j' \in \mathcal{J} : p_{j'} = ja} \gamma_{j'} \right\} \quad \forall j \in \mathcal{J},$$

and each $w_{ja} \geq 1$ ($ja \in \Sigma$) is defined recursively as

$$w_{ja} := \gamma_j - \sum_{j' \in \mathcal{J} : p_{j'} = ja} \gamma_{j'} \quad \forall ja \in \Sigma.$$
In particular, remember that $\mathcal{J}$ denotes the set of decision points, and given any $j \in \mathcal{J}$, $A_j$ is the set of actions available at $j$. The parent sequence of $j$, denoted $p_j$, is the last sequence (decision point-action pair) on the path from the root of the tree-form decision process to $j$.

**Theorem 3.1.** The dilatable global entropy function $\tilde{\phi} : Q \rightarrow \mathbb{R}_{\geq 0}$ is a regularizer for the sequence-form polytope $Q$. It is 1-strongly convex with respect to the $\ell_2$ (Euclidean) norm, and $(1/\|Q\|_1)$-strongly convex with respect to the $\ell_1$ norm, where $\|Q\|_1$ is the $\ell_1$-diameter of $Q$, that is, $\|Q\|_1 := \max_{q \in Q} \|q\|_1$.

**References**


