15-888 Computational Game Solving (Fall 2021) Tue, September 28th 2021

Lecture 7

Optimistic/predictive regret minimization via online optimization

Instructor: Gabriele Farina*

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 $\boldsymbol{x}^t \in \mathcal{X} \subseteq \mathbb{R}^n$;
- ObserveLoss($\boldsymbol{\ell}^t$) receives a *utility vector* $\boldsymbol{\ell}^t$ that is meant to evaluate the strategy $\boldsymbol{x}^t$ 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 $\boldsymbol{\ell}^t$ 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 $\mathcal{X}$ be a set. A *predictive regret minimizer for* $\mathcal{X}$ is an algorithm that interacts with the environment through two operations:

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

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

$$R^T := \max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \sum_{t=1}^{T} (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{x}} - (\boldsymbol{\ell}^t)^\top \boldsymbol{x}^t \right\}. \tag{1}$$

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

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

*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 $\boldsymbol{m}^t$ 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,

$$R^T \leq \alpha + \beta \sum_{t=1}^{T} \|\boldsymbol{\ell}^t - \boldsymbol{m}^t\|_*^2 - \gamma \sum_{t=2}^{T} \|\boldsymbol{x}^t - \boldsymbol{x}^{t-1}\|^2. \tag{2}$$

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

$$\max_{\boldsymbol{x} \in \mathcal{X}} \min_{\boldsymbol{y} \in \mathcal{Y}} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y} \tag{3}$$

is to use *self play*. Back then, we instantiated two regret minimization algorithms, $\mathcal{R}_\mathcal{X}$ and $\mathcal{R}_\mathcal{Y}$, for the domains of the maximization and minimization problem, respectively. At each time t the two regret minimizers output strategies $\boldsymbol{x}^t$ and $\boldsymbol{y}^t$, respectively. Then, they receive as feedback the vectors $\boldsymbol{\ell}_\mathcal{X}^t, \boldsymbol{\ell}_\mathcal{Y}^t$ defined as

$$\boldsymbol{\ell}_\mathcal{X}^t := \boldsymbol{A}\boldsymbol{y}^t, \qquad \boldsymbol{\ell}_\mathcal{Y}^t := -\boldsymbol{A}^\top \boldsymbol{x}^t, \tag{4}$$

where $\boldsymbol{A}$ is Player 1's payoff matrix (see also Figure 1).

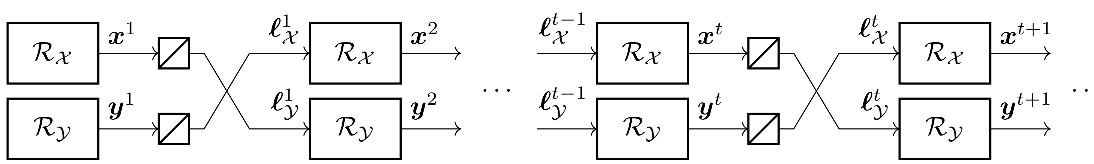

Figure 1: The flow of strategies and utilities in regret minimization for games. The symbol ⍁ denotes computation/construction of the utility vector.

We then proved that the average strategies

$$\bar{\boldsymbol{x}}^T := \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}^t, \qquad \bar{\boldsymbol{y}}^T := \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{y}^t$$

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

$$\gamma(\bar{\boldsymbol{x}}^T, \bar{\boldsymbol{y}}^T) \leq \frac{R_\mathcal{X}^T + R_\mathcal{Y}^T}{T}.$$

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,

$$\boldsymbol{m}_{\mathcal{X}}^t \coloneqq \boldsymbol{\ell}_{\mathcal{X}}^{t-1}, \qquad \boldsymbol{m}_{\mathcal{Y}}^t \coloneqq \boldsymbol{\ell}_{\mathcal{Y}}^{t-1}.$$

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

$$\begin{aligned}
\gamma(\bar{\boldsymbol{x}}^T, \bar{\boldsymbol{y}}^T) &\le \frac{R_{\mathcal{X}}^T + R_{\mathcal{Y}}^T}{T} \\
&\le \frac{1}{T}\left(\alpha + \beta \sum_{t=1}^T \left\|\boldsymbol{A}\boldsymbol{y}^t - \boldsymbol{A}\boldsymbol{y}^{t-1}\right\|_*^2 - \gamma \sum_{t=2}^T \left\|\boldsymbol{x}^t - \boldsymbol{x}^{t-1}\right\|^2\right) \\
&\qquad\qquad + \frac{1}{T}\left(\alpha + \beta \sum_{t=1}^T \left\|\boldsymbol{A}^\top\boldsymbol{x}^t - \boldsymbol{A}^\top\boldsymbol{x}^{t-1}\right\|_*^2 - \gamma \sum_{t=2}^T \left\|\boldsymbol{y}^t - \boldsymbol{y}^{t-1}\right\|^2\right) \\
&\le \frac{2\alpha}{T} + \frac{\beta\|\boldsymbol{A}\|_{\mathrm{op}}^2 - \gamma}{T} \sum_{t=1}^T \|\boldsymbol{y}^t - \boldsymbol{y}^{t-1}\|^2 + \frac{\beta\|\boldsymbol{A}\|_{\mathrm{op}}^2 - \gamma}{T} \sum_{t=1}^T \|\boldsymbol{x}^t - \boldsymbol{x}^{t-1}\|^2, \qquad (5)
\end{aligned}$$

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

Inequality (5) immediately implies that if $\|\boldsymbol{A}\|_{\mathrm{op}}^2 \le \gamma/\beta$ then the saddle point gap $\gamma(\bar{\boldsymbol{x}}^T, \bar{\boldsymbol{y}}^T)$ vanishes at the rate $O(1/T)$. Since rescaling $\boldsymbol{A}$ in (3) does not change the set of saddle points, we can always rescale $\boldsymbol{A}$ 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 $\boldsymbol{m}^t$ is set to the $\mathbf{0}$ vector at all t. In both algorithm, $\eta > 0$ is an arbitrary step size parameter, $\mathcal{X} \subseteq \mathbb{R}^n$ is a convex and compact set, and $\varphi : \mathcal{X} \to \mathbb{R}_{\ge 0}$ 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 φ, a standard surrogate notion of distance in convex optimization, defined as

$$D_\varphi(\boldsymbol{x} \,\|\, \boldsymbol{c}) \coloneqq \varphi(\boldsymbol{x}) - \varphi(\boldsymbol{c}) - \nabla\varphi(\boldsymbol{c})^\top(\boldsymbol{x} - \boldsymbol{c}) \qquad \forall\, \boldsymbol{x}, \boldsymbol{c} \in \mathcal{X}.$$

Remark 2.1. The divergence $D_\varphi(\boldsymbol{x} \,\|\, \boldsymbol{c})$ is a generalization of the typical notion of distance between $\boldsymbol{x}$ and $\boldsymbol{c}$. When the regularizer φ 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

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

function NextStrategy($\boldsymbol{m}^t$)
    [$\triangleright$ Set $\boldsymbol{m}^t = \boldsymbol{0}$ for non-predictive version]
    return $\operatorname*{arg\,max}_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ (\boldsymbol{L}^{t-1} + \boldsymbol{m}^t)^\top \hat{\boldsymbol{x}} - \frac{1}{\eta}\varphi(\hat{\boldsymbol{x}}) \right\}$

function ObserveUtility($\boldsymbol{\ell}^t$)
    $\boldsymbol{L}^t \leftarrow \boldsymbol{L}^{t-1} + \boldsymbol{\ell}^t$

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

2 **function** NextStrategy($\boldsymbol{m}^t$)
    [$\triangleright$ Set $\boldsymbol{m}^t = \boldsymbol{0}$ for non-predictive version]
3   **return** $\operatorname*{arg\,max}_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ (\boldsymbol{m}^t)^\top \hat{\boldsymbol{x}} - \frac{1}{\eta} D_\varphi(\hat{\boldsymbol{x}} \,\|\, \boldsymbol{z}^{t-1}) \right\}$

4 **function** ObserveUtility($\boldsymbol{\ell}^t$)
5     $\boldsymbol{z}^t \leftarrow \operatorname*{arg\,max}_{\hat{\boldsymbol{z}} \in \mathcal{X}} \left\{ (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{z}} - \frac{1}{\eta} D_\varphi(\hat{\boldsymbol{z}} \,\|\, \boldsymbol{z}^{t-1}) \right\}$

Euclidean norm), then

$$\begin{aligned} D_\varphi(\boldsymbol{x} \,\|\, \boldsymbol{c}) &= \varphi(\boldsymbol{x}) - \varphi(\boldsymbol{c}) - \nabla\varphi(\boldsymbol{c})^\top(\boldsymbol{x} - \boldsymbol{c})\rangle \\ &= \frac{1}{2}\|\boldsymbol{x}\|_2^2 - \frac{1}{2}\|\boldsymbol{c}\|_2^2 - \boldsymbol{c}^\top(\boldsymbol{x} - \boldsymbol{c}) \\ &= \frac{1}{2}\|\boldsymbol{x}\|_2^2 + \frac{1}{2}\|\boldsymbol{c}\|_2^2 - \boldsymbol{c}^\top\boldsymbol{x} \\ &= \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{c}\|_2^2 \end{aligned}$$

and so we recover that $D_\varphi(\boldsymbol{x} \,\|\, \boldsymbol{c})$ is half of the squared Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{c}$.

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

$$\begin{aligned} \operatorname*{arg\,max}_{\hat{\boldsymbol{z}} \in \mathcal{X}} \left\{ (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{z}} - \frac{1}{2\eta} D_\varphi(\hat{\boldsymbol{z}} \,\|\, \boldsymbol{z}^{t-1}) \right\} &= \operatorname*{arg\,max}_{\hat{\boldsymbol{z}} \in \mathcal{X}} \left\{ (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{z}} - \frac{1}{2\eta} \|\hat{\boldsymbol{z}} - \boldsymbol{z}^{t-1}\|_2^2 \right\} \\ &= \operatorname*{arg\,max}_{\hat{\boldsymbol{z}} \in \mathcal{X}} \left\{ (\eta\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{z}} - \frac{1}{2} \|\hat{\boldsymbol{z}} - \boldsymbol{z}^{t-1}\|_2^2 \right\} \\ &= \operatorname*{arg\,max}_{\hat{\boldsymbol{z}} \in \mathcal{X}} \left\{ (\boldsymbol{z}^{t-1} + \eta\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{z}} - \frac{1}{2} \|\hat{\boldsymbol{z}}\|_2^2 \right\} \\ &= \operatorname*{arg\,max}_{\hat{\boldsymbol{z}} \in \mathcal{X}} \left\{ \frac{1}{2} \left\|\hat{\boldsymbol{z}} - (\boldsymbol{z}^{t-1} + \eta\boldsymbol{\ell}^t)\right\|_2^2 \right\} \\ &= \mathrm{Proj}_{\mathcal{X}}(\boldsymbol{z}^{t-1} + \eta\boldsymbol{\ell}^t). \end{aligned}$$

Remark 2.3. In the non-predictive version of OMD, $\boldsymbol{m}^t = \boldsymbol{0}$ at all times t. In that case, the proximal step on Line 3 in Algorithm 2 reduces to

$$\boldsymbol{x}^t = \arg\max_{\hat{\boldsymbol{x}}\in\mathcal{X}} \left\{ -\frac{1}{\eta} D_\varphi(\hat{\boldsymbol{x}} \,\|\, \boldsymbol{z}^{t-1}) \right\} = \arg\min_{\hat{\boldsymbol{x}}\in\mathcal{X}} D_\varphi(\hat{\boldsymbol{x}} \,\|\, \boldsymbol{z}^{t-1}) = \boldsymbol{z}^{t-1}.$$

2.1 Regret guarantee

Predictive FTRL and predictive OMD satisfy the following regret bound.

Proposition 2.1. Let Ω denote the range of φ over $\mathcal{X}$, that is, $\Omega \coloneqq \max_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}}\{\varphi(\boldsymbol{x}) - \varphi(\boldsymbol{x}')\}$. At all times T, the regret cumulated by predictive FTRL (Algorithm 1) and predictive OMD (Algorithm 2) compared to *any* strategy $\hat{\boldsymbol{x}} \in \mathcal{X}$ is bounded as

$$R^T \le \frac{\Omega}{\eta} + \eta \sum_{t=1}^{T} \|\boldsymbol{\ell}^t - \boldsymbol{m}^t\|_*^2 - \frac{1}{c\eta} \sum_{t=2}^{T} \|\boldsymbol{x}^t - \boldsymbol{x}^{t-1}\|^2,$$

where $c = 4$ for FTRL and $c = 8$ for OMD, and where $\|\cdot\|_*$ denotes the dual of the norm $\|\cdot\|$ with respect to which φ 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 φ 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 φ 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 φ 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* $\nabla\varphi(\boldsymbol{x})$ of d at any point $\boldsymbol{x} \in \mathcal{X}$; and

- the gradient of the convex conjugate φ^* of φ at any point $\boldsymbol{g} \in \mathbb{R}^n$:

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

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

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

For Line 3 in predictive OMD, we have

$$\begin{aligned}
\arg\max_{\hat{\boldsymbol{x}}\in\mathcal{X}} \left\{(\boldsymbol{m}^t)^\top \hat{\boldsymbol{x}} - \frac{1}{\eta} D_\varphi(\hat{\boldsymbol{x}} \,\|\, \boldsymbol{z}^{t-1})\right\} &= \arg\max_{\hat{\boldsymbol{x}}\in\mathcal{X}} \big\{(\eta \boldsymbol{m}^t)^\top \hat{\boldsymbol{x}} - D_\varphi(\hat{\boldsymbol{x}} \,\|\, \boldsymbol{z}^{t-1})\big\} \\
&= \arg\max_{\hat{\boldsymbol{x}}\in\mathcal{X}} \big\{(\eta \boldsymbol{m}^t)^\top \hat{\boldsymbol{x}} - \varphi(\hat{\boldsymbol{x}}) + \nabla\varphi(\boldsymbol{z}^{t-1})^\top \hat{\boldsymbol{x}}\big\} \\
&= \arg\max_{\hat{\boldsymbol{x}}\in\mathcal{X}} \big\{(\eta \boldsymbol{m}^t + \nabla\varphi(\boldsymbol{z}^{t-1}))^\top \hat{\boldsymbol{x}} - \varphi(\hat{\boldsymbol{x}})\big\} \\
&= \nabla\varphi^*(\eta \boldsymbol{m}^t + \nabla\varphi(\boldsymbol{z}^{t-1})),
\end{aligned}$$

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

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

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

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

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.[a] The *dilatable global entropy distance generating function* $\tilde{\varphi}$ is the function $\tilde{\varphi} : Q \to \mathbb{R}_{\geq 0}$ defined as

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

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

$$\gamma_j \coloneqq 1 + \max_{a\in A_j} \left\{ \sum_{j'\in\mathcal{J}: p_{j'}=ja} \gamma_{j'} \right\} \qquad \forall\, j \in \mathcal{J}, \tag{6}$$

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

$$w_{ja} \coloneqq \gamma_j - \sum_{j'\in\mathcal{J}: p_{j'}=ja} \gamma_{j'} \qquad \forall\, ja \in \Sigma.$$

[a] 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 $\tilde{\varphi} : Q \to \mathbb{R}_{\geq 0}$ is a regularizer for the sequence-form polytope Q. It is 1-strongly convex with respect to the ℓ_2 (Euclidean) norm, and $(1/\|Q\|_1)$-strongly convex with respect to the ℓ_1 norm, where $\|Q\|_1$ is the ℓ_1-diameter of Q, that is, $\|Q\|_1 \coloneqq \max_{\boldsymbol{q} \in Q} \|\boldsymbol{q}\|_1$.

References

Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. In *Proceedings of the Conference on Learning Theory (COLT)*, 2012.

Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. In *Proceedings of the Conference on Learning Theory (COLT)*, 2013a.

Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2013b.

Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire. Fast convergence of regularized learning in games. In *Advances in Neural Information Processing Systems*, 2015.

Shai Shalev-Shwartz and Yoram Singer. A primal-dual perspective of online learning algorithms. *Machine Learning*, 69(2-3):115–142, 2007.

Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Better regularization for sequential decision spaces: Fast convergence rates for Nash, correlated, and team equilibria. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2021.