Multi-Agent Learning: From Theory to Practice

Report of "No-regret learning in convex games"

Andrea Vitobello

May 30th, 2023

1 Summary

1.1 Introduction

This paper discusses regret minimization in general convex games, introducing regret types, as well as their respective equilibrium, comprised inside the spectrum of Coarse Correlated Equilibrium (CCE) and Correlated Equilibrium (CE). This paper furthermore displays a general algorithm to minimize these regrets.

1.2 Setting

The setting of this paper is repeated general-sum multi-agent convex games, where at every round t each agent i can choose an action from a feasible convex and compact region $A_i \in \mathbb{R}^d$ and observes a loss $l_t(a_t)$. We consider an online scenario where agents only know their own feasible region and observe their own payoff structure, which however depends also on the actions performed by other players $a_{\neg i}$: it is thus necessary for agents to learn a course of action as it is unfeasible to calculate and coordinate with other players to reach an equilibrium.

We define the goal of a learning agent as regret minimization. Stoltz and Lugosi [2] defined a continuum of regret measures called Φ-regret, where Φ is a collection of action transformations $\phi : A \rightarrow A$. Different choices of Φ are related to different types of regret and equilibrium. An agent calculates its Φ-regret by comparing the loss it has obtained so far with the minimum loss it would have obtained by transforming his actions according to some ϕ in Φ.

$$\rho_t^{\Phi} = \sup_{\phi \in \Phi} \sum_{t=1}^{T} (l_t(a_t)) - l_t(\phi(a_t)))$$

If an algorithm achieves no Φ-regret it means that the average Φ-regret per trial eventually falls below some ϵ. In other words

$$\sum_{t=1}^{T}(l_t(a_t)) \leq \sum_{t=1}^{T}(l_t(\phi(a_t)) + g(T, A, L, \Phi)$$

where g is a function sub-linear in T for any fixed A, L, Φ, and L is a set of convex loss functions. The most relevant regret types in this paper are the following, each with its respective equilibrium stronger than the previous one:

- external regret (Φ_{EXT}-regret): regret relative to all constant transformations which, given any action $a_i \in A_i$ as input, always output a fixed action $\phi(a_i) = \hat{a}_i$. If no external regret is achieved, then the solution reached corresponds to a CCE. Notice that an agent that reaches no external regret might still have an incentive to deviate, and this is the reason why CCE is not a sturdy enough solution concept in repeated games.
- extensive-form regret (Φ_{EF}-regret): regret relative to extensive-form games and which leads to Extensive-Form Correlated Equilibrium (EFCE) when no-extensive-form regret is achieved.
- linear regret (Φ_{LIN}-regret): regret relative to all linear transformations that map the feasible region A into itself.
- finite element regret (Φ_{FE}-regret): it covers all transformations from A into itself when A is a polyhedral feasible region.
- swap regret (Φ_{SWAP}-regret): regret relative to transformation sets Φ that contain all possible functions ϕ. The corresponding equilibrium is the CE, and thus it is the strongest possible equilibrium discussed in the paper.

1.3 Regret minimizer: groundings

The regret minimizer is the core of the paper. Such algorithm, which can be found in the last paragraph of this section, is defined for general sets of transformations Φ and it assumes the existence of two subroutines $\mathcal{A}'$, $\mathcal{A}''$.

$\mathcal{A}'$ is a subroutine that computes for each round t an approximate fixed point, according to an arbitrary norm $||.||_A$ and with respect to a transformation ϕ_t. In particular, the approximation is up to a parameter ϵ_t that is sub-linear in the round number t and that represents the regret bound.

$\mathcal{A}''$ is a subroutine calculating the next round's ϕ_{t+1} through an external regret minimizer on Φ given the loss function observed at round t.

Notice that the algorithm is thus general for any transformation set Φ as it does not depend on any particular property of Φ: indeed the algorithm accesses the transformation set only through the two subroutines. Moreover this means that the algorithm is deterministic as long as the subroutines are.

To prove the convergence of the algorithm to no-regret, the space of transformations Φ is embedded into a vector space such that each $\phi \in \Phi$ can be written as a d-tuple of coordinate functions $(\psi_1, ..., \psi_d)$, where $\psi_i : A \rightarrow \mathbb{R}$ is a member of a Reproducing-Kernel Hilbert Space[1] (RKHS) $H \subset A \rightarrow \mathbb{R}$. We then assume that Φ is a convex and compact subset of H^d. The successive step consists in computing suitable components ψ_i and subroutines for the transformation set we are considering when minimizing regret. All these ingredients directly depend on the specific set of transformations under evaluation.

Thanks to the reproducing-kernel property, in the specific case of finite-dimensional Φ it is possible to manually[2] use the *kernel trick* [3] to compute $\psi_i(a)$ as $\langle \psi_i, K(a) \rangle$ for some fixed and possibly-non-linear function $K(\cdot)$. Lastly, by restricting the dimension of H such that it is isomorphic to $\mathbb{R}^p$ for some finite p, it is possible to define each Φ as a set of $d \times p$ matrices, one for each transformation ϕ, along with a kernel function $K : A \rightarrow \mathbb{R}^p$ satisfying $\phi K(a) \in A$ for all $a \in A$ and for all $\phi \in \Phi$. We now instantiate the algorithm for specific transformations sets and their respective regret.

- Linear Regret: Φ_{LIN} is the set of linear transformations that map A into itself and it is a set of square $d \times d$ matrices, thus we can take K as the identity. $\mathcal{A}'$ finds a fixed point by selecting an eigenvector of $\phi - I$, requiring time polynomial in d, while $\mathcal{A}''$ can achieve no external regret on Φ through GIGA or lazy projection [4].

- Finite Element Regret: finite-element transformations only apply to polyhedral feasible regions as each point of A is mapped over the p corners of A. In order to do so, it is necessary to triangulate A by dividing it into mutually exclusive and exhaustive d-simplices so that each point is strictly defined by weighting the corners of the simplex it falls into. K is thus defined to be the mapping from a polyhedral feasible set to the barycentric coordinate space of dimension p such that each one of the p corners of A is associated with a dimension in $\mathbb{R}^p$. $\mathcal{A}'$ is implemented by individually checking every d-simplex separately, for a cost of $O(d^3)$ each. $\mathcal{A}''$ is implemented by running any no external regret in Φ for each corner of A, with an expected cost of $O(pd^3)$.

- Extensive-Form Regret: assuming T as a player's sequence tree, Φ_{EF} is defined to be the set of all transformations that, given as input a sequence weight vector [10] on T representing the behaviour of the player, they give back another possibly different sequence weight vector on T.

Notice that the above results also hold when players can only play corners of polyhedral games or pure actions in matrix games: what is fundamental is that the empirical distribution of moves in repeated rounds is randomized and follows the same distribution prescribed by the equilibrium.

[1] A Hilbert space is an inner product space. A RKHS is a Hilbert space of functions where the evaluation at each point is a continuous linear function.

[2] The authors of the paper manually produce the kernel function according to the transformation sets, thus a limited-size Φ is fundamental to make the process feasible.

1.4 Regret Minimizer: algorithm

Given a feasible region A, transformation set Φ, initial transformation ϕ_1 and subroutines $\mathcal{A}'$, $\mathcal{A}''$; for $t = 1, ..., T$:

1. To find an approximate fixed point to play, send transformation ϕ_t and accuracy parameter $\epsilon_t = 1/\sqrt{t}$ to $\mathcal{A}'$; receive as output a_t satisfying $||\phi_t(a_t) - a_t||_A \leq \epsilon_t$.
2. Play a_t; observe a loss function l_t; incur in loss $l_t(a_t)$.
3. Define a transformation loss function $m_t : \Phi \to \mathbb{R}$ as $m_t(\phi) = l_t(\phi(a_t))$.
4. Send m_t to $\mathcal{A}''$; receive transformation $\phi_{t+1} \in \Phi$.

2 Technical Evaluation

This paper is relatively poor in proofs and experiments. The only proof revolves around the demonstration that the regret minimizer indeed achieves no Φ-regret. The necessary ingredients are:

- a convex and compact feasible region A such that $A_i \in \mathbb{R}^d$ for all i[3] and $||a||_A \leq C_1$ for some constant $C_1 \geq 0$;
- a set of convex loss functions L with bounded sub-gradients: $||\delta l(a)||_{A^*} \leq C_2$ for any $a \in A$, any $l \in L$ and some constant $C_2 \geq 0$;
- a set of transformations $\Phi \subset H^d$, where $H \subset A \to \mathbb{R}$ is a RKHS;
- an algorithm $\mathcal{A}'$ to compute ϵ-approximate fixed points of any $\phi \in \Phi$;
- an algorithm $\mathcal{A}''$ which achieves no external regret on Φ for any set of loss functions with bounded sub-gradients.

Thanks to the detailed and effective preliminary explanations, the proof of the theorem is straightforward and well-developed, although it would have benefited from an additional intermediate step better elucidating how Hölder's inequality comes into play. Moreover, the achieved result is remarkably general, in accordance with the purpose of the paper.

In fact, we see that all the assumptions are relatively general and not particularly restrictive; the only mild constraints are those on the embedding of Φ and on the existence of $\mathcal{A}'$ and $\mathcal{A}''$, which might be hard to produce depending on the specific set of transformations we take into consideration; this topic is further developed in the following section, however we immediately point out that the authors of the paper already suggest the characterization of some embeddings and subroutines, thus alleviating some implementation troubles for the paper readers. More specifically, the authors suggest possible configurations of both subroutines for Φ_{LIN} and Φ_{FE}, while they do not allude to any subroutine in the case of Φ_{EF}, as it was noticed also by Bai et al., [9].

[3] Notice that $||.||_A$ is an arbitrary norm on $\mathbb{R}^d$ and $||.||_{A^*}$ is its dual.

More generally speaking, the crux of the paper is clearly identified and well presented. However, what is most debatable is the poor characterization of some concepts. It is most certainly impossible to write a completely self-contained paper as it would require to include an unbearable amount of explanations and topics; on the other hand, a good rule of thumb should be discussing extensively enough every notion that is preparatory for the main core of the paper, as well as validating as many decisions as possible. Some examples of poorly characterized notions are those of barycentric coordinate space and the rationale behind the specific transformation regarding no Φ_{FE}-regret, as well as the delineation of the extensive-form transformation. An immediate solution to this problem could have been an appendix in an extended version of the paper comprising a more detailed introduction of the fundamental concepts, possibly adorned with thorough and meticulous examples. The third section of the paper is particularly damaged by this lack of clarity.

Some additional trivialities are the occasional lack of a well-defined and rigorous nomenclature, such as in the absence of distinction between *choice node* and *choices* when discussing the sequence tree, as well as some arbitrariness, such as when discussing the run-time of $\mathcal{A}''$ in case of Finite-Element Regret.

3 Advantages and Drawbacks

What is most notable and praiseworthy about this paper and the approach it suggests is how general they are despite tackling the huge scope of general-sum online convex programming. Both the proof and the algorithm are valid for general transformation sets Φ as long as we can produce an embedding $(\psi_1, ..., \psi_d)$ and the necessary subroutines $\mathcal{A}'$, $\mathcal{A}''$. As a matter of fact, the framework introduced in this paper is still used nowadays, for example by Farina, et all., [5] with respect to EFCE, by Fujii [6] with respect to linear regret, and by Anagnostides et al., [7].

Since this approach is still used, we can immediately conclude that it is computationally efficient. In fact, the witty decoupling of the algorithm in subroutines $\mathcal{A}'$ and $\mathcal{A}''$ allows to split the effort of producing a no-regret minimizer into subtasks which may be more nimbly dealt with and optimized, such as in [7]. At the same time, one of the disadvantages of this approach, as noticed by Dudik and Gordon himself [8], is that the algorithm might still delegate potentially intractable sub-problems to an outside oracle. In particular, in order for the algorithm to work correctly, it is still necessary to produce the subroutines and an effective embedding of Φ, including the kernel function K, which might not always be straightforward tasks. In support of this claim, despite proving the validity of the results for general Φ, the authors of the paper only develop cases where Φ is finite dimensional and the dimension of H is manageable so that transformations can be directly worked with.

One additional drawback concerns the hazy definition of Φ_{FE}: instead of declaring the kernel function in order to fit Φ_{FE}, this set of transformations is actually defined by means of function K. This implies that Φ_{FE} is less portable to other scenarios than it could have been, and as a demonstration of this fact, one can find no reference about finite-element regret in the literature up to this date.

Lastly, a very slight drawback concerns the rigidity of this approach with respect to solution concepts that do not fall inside the Φ-regret spectrum, such as the Nash Equilibrium (NE). Such solutions cannot be computed with the proposed algorithm, but we deem this drawback as not particularly influential because in the setting of general-sum n-players games converging to a NE is extremely expensive and might yield payoffs arbitrarily worse than a CE [5]. Another result discouraging the use NE is that fully mixed NE are not asymptotically stable under Follow The Regularized Leader dynamics [11], and conversely also in mirror descent dynamics [12]: only strict NE[4] can be asymptotically stable in such scenarios.

4 Improvements and Extensions

First of all, it should be noted that every computational improvement concerning the subroutines $\mathcal{A}'$ and $\mathcal{A}''$ with respect to some transformation set Φ directly implies an improvement also in the overall algorithm for that same Φ, thus any such enhancement is well received. Moreover, since the algorithm and the demonstrated results focus on general applicability, any further extension to the framework is a laborious task.

As it was mentioned before, this paper demonstrates the validity of no-regret for general Φ, but eventually only studies cases where Φ is finite dimensional and H is manageable, so that the kernel mapping is tractable by directly working with the transformation matrices. A possible extension of this method would therefore involve high dimensional transformations. As the authors of this paper suggest, an attainable approach would be to derive kernelized no-Φ-regret algorithms where the specific transformations are not directly written but instead they are derived in terms of observed actions and loss functions. Moreover, the kernel trick allows to deal even with an infinite dimensional space [3], so this extension is quite interesting.

Lastly, it has been noted above that the approach proposed in the paper is general only with respect to equilibria that can be traced back to Φ-regret. On the other hand, it might be useful to employ different solution concepts, such as the Stackelberg equilibrium, which indeed can be learned through no-regret [13, 14], or the NE, possibly under some strict assumptions to enable a feasible computation, for example in the case of zero-sum games. Notice, however, that even though these solution concepts can be proven to be effective in some scenarios [15, 13], further generalizing the framework would require a notable effort which might outweigh the obtained results.

[4]A NE is said to be strict if every player has a unique best response.

References

[1] Gordon, Greenwald, Marks: *No-Regret Learning in Convex Games*, 2008.
[2] Stoltz, Lugosi: *Learning Correlated Equilibria in Games With Compact Sets of Strategies*, 2007.
[3] Cortes, Vapnik: *Support-Vector Networks*, 1995.
[4] Zinkevich: *Online Convex Programming and Generalized Infinitesimal Gradient Ascent*, 2003.
[5] Celli, Marchesi, Farina, Gatti: *Simple Uncoupled No-Regret Learning Dynamics for Extensive-Form Correlated Equilibrium*, 2021.
[6] Fujii: *Bayes Correlated Equilibria and No-Regret Dynamics*, 2023.
[7] Anagnostides, Daskalakis, Farina, Fishelson, Golowich, Sandholm: *Near-Optimal No-Regret Learning for Correlated Equilibria in Multi-Player General-Sum Games*, 2023.
[8] Dudik, Gordon: *A Sampling-Based Approach to Computing Equilibria in Succint Extensive-Form Games*, 2009.
[9] Bai, Jin, Mei, Song, Yu: *Efficient Phi-Regret Minimization in Extensive-Form Games via Online Mirror Descent*, 2022.
[10] Forges, von Stengel: *Computationally efficient coordination in game trees*, 2002.
[11] Flokas, Vlatakis-Gkaragkounis, Lianeas, Mertikopoulos, Piliouras: *No-Regret Learning and Mixed Nash Equilibria: They Do Not Mix*, 2020.
[12] McMahan: *Follow-the-Regularized-Leader and Mirror Descent: Equivalence Theorems and L1 Regularization*, 2011.
[13] Letchford, Conitzer, Munagala: *Learning and Approximating the Optimal Strategy to Commit To*, 2009.
[14] Balcan, Blum, Haghtalab Procaccia: *Commitment Without Regrets: Online Learning in Stackelberg Security Games*, 2015.
[15] Blum, Even-Dar, Ligett: *Routing Without Regret: On Convergence to Nash Equilibria of Regret-Minimizing Algorithms in Routing Games*, 2010.