
Convex Repeated Games and Fenchel Duality

Shai Shalev-Shwartz

School of Computer Sci. & Eng.,
The Hebrew University, Jerusalem 91904, Israel shais@cs.huji.ac.il

Yoram Singer

Google Inc.
1600 Amphitheater Parkway, Mountain View, CA 94043, USA
singer@google.com

Abstract

We describe and analyze an algorithmic framework for playing convex repeated games. In each trial of the repeated game, the first player predicts a vector and then the second player responds with a loss function over the vector. Based on a generalization of Fenchel duality, we derive an algorithmic framework for the first player and analyze the player's regret. We then use our algorithmic framework and its corresponding regret analysis for online learning problems and for boosting.

1 Introduction

Many problems arising in machine learning, such as online learning and boosting, can be modeled as a convex repeated game. A convex repeated game is a two players game which is performed in a sequence of consecutive trials. We study this game from the view point of the first player, which we term the learner and refer to the second player as the environment. At each trial of the game, the learner is required to predict a vector from some domain and then, the environment responds with a loss function over the domain. The learner then suffers a loss according to the assessment of the loss function on the vector he predicts. The goal of the learner is to minimize the cumulative loss it suffers along its run.

In this paper we describe and analyze a general algorithmic framework for playing convex repeated games. Our framework is based on casting regret bounds as optimization problems. A regret bound compares the cumulative loss suffered by the learner to the cumulative loss of any competing fixed vector. The competing vector can be chosen in hindsight after observing the entire sequence of loss functions. Regret bounds are universal in the sense that they hold for any possible competing vector in a given set of admissible vectors. We therefore cast the universal regret bound as an optimization problem. The best competing vector, which can only be determined in hindsight, is the minimizer of the optimization problem. Generalizing the notion of Fenchel duality, we derive a dual optimization problem, which can be optimized incrementally, as the game proceeds. In order to derive explicit quantitative regret bounds we make an immediate use of the fact that dual objective lower bounds the primal objective. We therefore reduce the process of playing convex repeated games to the task of incrementally increasing the dual objective function. The amount by which the dual increases serves as a new and natural notion of progress. By doing so we are able to tie the primal objective value, the cumulative loss of the learner, and the increase in the dual.

After establishing our notation and pointing to a few mathematical tools used throughout the paper (Sec. 2), we formally define convex repeated games (Sec. 3). Our main tool for deriving algorithms for playing convex repeated games is a generalization of Fenchel duality, described in Sec. 4. Our algorithmic framework is given in Sec. 5 and analyzed in Sec. 6. The generality of our framework

allows us to utilize it in different problems arising in machine learning. In Sec. 7 we underscore the applicability of our framework for online learning, and specifically, we derive online algorithms for complex decision problems. Next, in Sec. 8, we derive and analyze boosting algorithms from our framework. In particular, we give a new analysis for known boosting methods and also draw new directions for deriving novel boosting algorithms. We conclude with a discussion and point to related work in Sec. 9. Due to the lack of space, some of the details are omitted from the paper and are deferred to appendices, which are submitted in an accompanying paper.

2 Mathematical Background

In this section we establish our notation and also point to a few mathematical tools used throughout the paper. We denote scalars with lower case letters (e.g. x and ω), and vectors with bold face letters (e.g. \mathbf{x} and $\boldsymbol{\omega}$). The inner product between vectors \mathbf{x} and $\boldsymbol{\omega}$ is denoted by $\langle \mathbf{x}, \boldsymbol{\omega} \rangle$. Sets are designated by upper case letters (e.g. Ω). The set of non-negative real numbers is denoted by \mathbb{R}_+ . For any $k \geq 1$, the set of integers $\{1, \dots, k\}$ is denoted by $[k]$.

We next list basic definitions from convex analysis. The reader familiar with convex analysis may proceed to Lemma 1 while for a more thorough introduction see for example [1]. A set Ω is convex if for any two vectors $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2$ in Ω , all the line between $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ is also within Ω . That is, for any $\alpha \in [0, 1]$ we have that $\alpha\boldsymbol{\omega}_1 + (1 - \alpha)\boldsymbol{\omega}_2 \in \Omega$. A set Ω is open if every point in Ω has a neighborhood lying in Ω . A set Ω is closed if its complement is an open set. A function $\ell : \Omega \rightarrow \mathbb{R}$ is closed and convex if for any scalar $\alpha \in \mathbb{R}$, the level set $\{\boldsymbol{\omega} : \ell(\boldsymbol{\omega}) \leq \alpha\}$ is closed and convex. The Fenchel conjugate of a function $f : \Omega \rightarrow \mathbb{R}$ is defined as $f^*(\boldsymbol{\theta}) = \sup_{\boldsymbol{\omega} \in \Omega} \langle \boldsymbol{\omega}, \boldsymbol{\theta} \rangle - f(\boldsymbol{\omega})$. If f is closed and convex then the Fenchel conjugate of f^* is f itself. The Fenchel-Young inequality states that for any $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$ we have that $f(\boldsymbol{\omega}) + f^*(\boldsymbol{\theta}) \geq \langle \boldsymbol{\omega}, \boldsymbol{\theta} \rangle$. A vector $\boldsymbol{\lambda}$ is a sub-gradient of a function f at $\boldsymbol{\omega}$ if for all $\boldsymbol{\omega}' \in \Omega$ we have that $f(\boldsymbol{\omega}') - f(\boldsymbol{\omega}) \geq \langle \boldsymbol{\omega}' - \boldsymbol{\omega}, \boldsymbol{\lambda} \rangle$. The differential set of f at $\boldsymbol{\omega}$, denoted $\partial f(\boldsymbol{\omega})$, is the set of all sub-gradients of f at $\boldsymbol{\omega}$. Sub-gradients play an important role in the definition of Fenchel conjugate. In particular, the following lemma states that if $\boldsymbol{\lambda} \in \partial f(\boldsymbol{\omega})$ then Fenchel-Young inequality holds with equality.

Lemma 1 *Let f be a closed and convex function and let $\partial f(\boldsymbol{\omega}')$ be its differential set at $\boldsymbol{\omega}'$. Then, for all $\boldsymbol{\lambda}' \in \partial f(\boldsymbol{\omega}')$ we have, $f(\boldsymbol{\omega}') + f^*(\boldsymbol{\lambda}') = \langle \boldsymbol{\lambda}', \boldsymbol{\omega}' \rangle$.*

The proof is given in Appendix C. If f is differentiable at $\boldsymbol{\omega}$ then $\partial f(\boldsymbol{\omega})$ consists of a single vector which is called the gradient of f at $\boldsymbol{\omega}$ and is denoted by $\nabla f(\boldsymbol{\omega})$. Whenever f is twice differentiable we denote by $\nabla^2 f(\boldsymbol{\omega})$ the Hessian of f which is the matrix of second order derivatives of f with respect to the components of $\boldsymbol{\omega}$.

3 Convex Repeated Games

A convex repeated game is a two players game which is performed in a sequence of consecutive trials. We study this game from the view point of the first player, which we term the learner and refer to the second player as the environment. At trial t , the learner is required to predict a vector $\boldsymbol{\omega}_t \in \Omega$, where Ω is a convex set. After the prediction is made the environment presents a function $\ell_t : \Omega \rightarrow \mathbb{R}_+$, where ℓ_t is a convex and closed function. The learner then suffers a loss $\ell_t(\boldsymbol{\omega}_t)$. The goal of the learner is to minimize the cumulative loss it suffers along its run.

For any number of trials T and for any fixed $\boldsymbol{\omega} \in \Omega$, we define the *regret* of the learner for not playing $\boldsymbol{\omega}$ at the first T trials to be $\frac{1}{T} \sum_{t=1}^T \ell_t(\boldsymbol{\omega}_t) - \frac{1}{T} \sum_{t=1}^T \ell_t(\boldsymbol{\omega})$. The right-hand summand in the above expression would have been the average loss of the learner had she chosen to set $\boldsymbol{\omega}_t$ to be equal to $\boldsymbol{\omega}$ for all $t \in [T]$. Naturally, the problem of finding $\boldsymbol{\omega}$ which minimizes the right-hand summand above depends on the entire sequence of loss functions. The regret reflects the amount of excess loss the learner suffers for not knowing in advance the complete sequence of loss functions.

In this paper we provide a family of algorithms for convex repeated games which attain regret bounds of the form

$$\forall \boldsymbol{\omega} \in \Omega, \quad \frac{1}{T} \sum_{t=1}^T \ell_t(\boldsymbol{\omega}_t) - \frac{1}{T} \sum_{t=1}^T \ell_t(\boldsymbol{\omega}) \leq O\left(\frac{f(\boldsymbol{\omega})}{\sqrt{T}}\right), \quad (1)$$

where we refer to f as a complexity function over the set of admissible vectors Ω . Thus, these regret bounds imply that the regret of the online algorithm with respect to any vector within the set $\{\omega \in \Omega : f(\omega) = o(\sqrt{T})\}$ approaches zero as T goes to infinity. Our algorithmic framework is based on a generalization of Fenchel duality which we describe in the next section.

4 Generalized Fenchel Duality

Consider the following optimization problem,

$$\inf_{\omega \in \Omega} \left(f(\omega) + c \sum_{t=1}^T \ell_t(\omega) \right) ,$$

where c is a non-negative scalar. An equivalent problem is

$$\inf_{\omega_0, \omega_1, \dots, \omega_T} \left(f(\omega_0) + c \sum_{t=1}^T \ell_t(\omega_t) \right) \text{ s.t. } \omega_0 \in \Omega \text{ and } \forall t \in [T], \omega_t = \omega_0 .$$

Introducing T vectors $\lambda_1, \dots, \lambda_T$, each $\lambda_t \in \mathbb{R}^n$ is a vector of Lagrange multipliers for the equality constraint $\omega_t = \omega_0$, we obtain the following Lagrangian

$$\mathcal{L}(\omega_0, \omega_1, \dots, \omega_T, \lambda_1, \dots, \lambda_T) = f(\omega_0) + c \sum_{t=1}^T \ell_t(\omega_t) + \sum_{t=1}^T \langle \lambda_t, \omega_t - \omega_0 \rangle .$$

The dual problem is to maximize the dual objective function given as

$$\begin{aligned} \mathcal{D}(\lambda_1, \dots, \lambda_T) &= \inf_{\omega_0 \in \Omega, \omega_1, \dots, \omega_T} \mathcal{L}(\omega_0, \omega_1, \dots, \omega_T, \lambda_1, \dots, \lambda_T) \\ &= - \sup_{\omega_0 \in \Omega} \left(\langle \omega_0, \sum_{t=1}^T \lambda_t \rangle - f(\omega_0) \right) - c \sum_{t=1}^T \sup_{\omega_t} \left(\langle \omega_t, \frac{-\lambda_t}{c} \rangle - \ell_t(\omega_t) \right) \\ &= -f^* \left(\sum_{t=1}^T \lambda_t \right) - c \sum_{t=1}^T \ell_t^* \left(-\lambda_t/c \right) , \end{aligned}$$

where, following the exposition of Sec. 2, $f^*, \ell_1^*, \dots, \ell_T^*$ are the Fenchel conjugate functions of f, ℓ_1, \dots, ℓ_T . Therefore, the generalized Fenchel dual problem is

$$\sup_{\lambda_1, \dots, \lambda_T} -f^* \left(\sum_{t=1}^T \lambda_t \right) - c \sum_{t=1}^T \ell_t^* \left(-\lambda_t/c \right) . \quad (2)$$

Note that when $T = 1$ and $c = 1$, the above duality is the so called Fenchel duality.

5 A Template Learning Algorithm for Convex Repeated Games

In this section we describe a template learning algorithm for playing convex repeated game. Recall that we would like our learning algorithm to achieve a regret bound of the form given in Eq. (1). We start by rewriting Eq. (1) in a slightly different way. Let $c = \frac{1}{\sqrt{T}}$ and let U be a constant which does not depend on T . Then, Eq. (1) can be rewritten as

$$c \sum_{t=1}^T \ell_t(\omega_t) - U \leq \inf_{\omega \in \Omega} \left(f(\omega) + c \sum_{t=1}^m \ell_t(\omega) \right) . \quad (3)$$

Thus, up to constants, the cumulative loss of the learner lower bounds the optimum of the minimization problem on the right-hand side of Eq. (3). In the previous section we derived the generalized Fenchel dual of the right-hand side of Eq. (3). Our construction is based on the weak duality theorem stating that any value of the dual objective function is smaller than the optimum value of the primal problem. Our learning algorithm is therefore derived by incrementally ascending the dual objective function. Intuitively, by ascending the dual objective we move closer to the optimal primal value and therefore our performance becomes similar to the performance of the best fixed weight vector which minimizes the right-hand side of Eq. (3).

Initially, we use the elementary dual solution $\lambda_t^1 = \mathbf{0}$ for all t . We assume that $\inf_{\omega} f(\omega) = 0$ and for all $t \inf_{\omega} \ell_t(\omega) = 0$ which imply that $\mathcal{D}(\lambda_1^1, \dots, \lambda_T^1) = 0$. Assume in addition that the function f^* is differentiable. At trial t , the learner uses for prediction the vector

$$\omega_t = \nabla f^* \left(\sum_{i=1}^T \lambda_i^t \right) . \quad (4)$$

After predicting ω_t the learner receives the function ℓ_t and suffer the loss $\ell_t(\omega_t)$. Then the learner updates the dual variables as follows. Denote by ∂_t the differential set of ℓ_t at ω_t , that is,

$$\partial_t = \{ \lambda : \forall \omega \in \Omega, \ell_t(\omega) - \ell_t(\omega_t) \geq \langle \lambda, \omega - \omega_t \rangle \} . \quad (5)$$

The new dual variables $(\lambda_1^{t+1}, \dots, \lambda_T^{t+1})$ are set to be any set of vectors which satisfy the following two conditions:

$$\begin{aligned} (i). \quad & \exists \lambda' \in \partial_t \text{ s.t. } \mathcal{D}(\lambda_1^{t+1}, \dots, \lambda_T^{t+1}) \geq \mathcal{D}(\lambda_1^t, \dots, \lambda_{t-1}^t, -c\lambda', \lambda_{t+1}^t, \dots, \lambda_T^t) \\ (ii). \quad & \forall i > t, \lambda_i^{t+1} = \mathbf{0} \end{aligned} . \quad (6)$$

In the next section we show that condition (i) ensures that the increase of the dual at trial t is proportional to the loss $\ell_t(\omega_t)$. The second condition ensures that we can actually calculate the dual at trial t without any knowledge on the yet to be seen loss functions $\ell_{t+1}, \dots, \ell_T$.

We conclude this section with two update rules that trivially satisfies the above two conditions. The first update scheme simply finds $\lambda' \in \partial_t$ and set

$$\lambda_i^{t+1} = \begin{cases} -c\lambda' & \text{if } i = t \\ \lambda_i^t & \text{if } i \neq t \end{cases} . \quad (7)$$

The second update defines

$$(\lambda_1^{t+1}, \dots, \lambda_T^{t+1}) = \underset{\lambda_1, \dots, \lambda_T}{\operatorname{argmax}} \mathcal{D}(\lambda_1, \dots, \lambda_T) \quad \text{s.t.} \quad \forall i \neq t, \lambda_i = \lambda_i^t . \quad (8)$$

6 Analysis

In this section we analyze the performance of the template algorithm given in the previous section. Our proof technique is based on monitoring the value of the dual objective function. The main result is the following lemma which gives upper and lower bounds for the final value of the dual objective function.

Lemma 2 *Let f be a closed and convex function whose Fenchel dual f^* is twice differentiable and satisfies $f^*(\mathbf{0}) = 0$. Let ℓ_1, \dots, ℓ_T be a sequence of convex and closed functions whose Fenchel duals satisfy $\ell_t^*(\mathbf{0}) = 0$ for all $t \in [T]$. Suppose that a dual-incrementing algorithm which satisfies the conditions of Eq. (6) is run with f as a complexity function on the sequence ℓ_1, \dots, ℓ_T . Let $\omega_1, \dots, \omega_T$ be the sequence of primal vectors the algorithm generates and $\lambda_1^{T+1}, \dots, \lambda_T^{T+1}$ be its final sequence of dual variables. Then, there exists a sequence of vectors $\theta'_1, \dots, \theta'_T$ and a sequence of sub-gradients $\lambda'_1, \dots, \lambda'_T$, where $\lambda'_t \in \partial_t$ for all t , such that*

$$c \sum_{t=1}^T \ell_t(\omega_t) - \frac{c^2}{2} \sum_{t=1}^T \langle \lambda'_t, \nabla^2 f^*(\theta'_t) \lambda'_t \rangle \leq \mathcal{D}(\lambda_1^{T+1}, \dots, \lambda_T^{T+1}) \leq \inf_{\omega \in \Omega} f(\omega) + c \sum_{t=1}^T \ell_t(\omega) .$$

Proof The right inequality follows directly from the weak duality theorem. Turning to the left most inequality, denote $\Delta_t = \mathcal{D}(\lambda_1^{t+1}, \dots, \lambda_T^{t+1}) - \mathcal{D}(\lambda_1^t, \dots, \lambda_T^t)$ and note that $\mathcal{D}(\lambda_1^{T+1}, \dots, \lambda_T^{T+1})$ can be rewritten as

$$\mathcal{D}(\lambda_1^{T+1}, \dots, \lambda_T^{T+1}) = \sum_{t=1}^T \Delta_t - \mathcal{D}(\lambda_1^1, \dots, \lambda_T^1) = \sum_{t=1}^T \Delta_t , \quad (9)$$

where the last equality follows from the assumption that $f^*(\mathbf{0}) = \ell_1^*(\mathbf{0}) = \dots = \ell_T^*(\mathbf{0}) = 0$. The definition of the update implies that $\Delta_t \geq \mathcal{D}(\lambda_1^t, \dots, \lambda_{t-1}^t, -c\lambda'_t, \mathbf{0}, \dots, \mathbf{0}) - \mathcal{D}(\lambda_1^t, \dots, \lambda_{t-1}^t, \mathbf{0}, \mathbf{0}, \dots, \mathbf{0})$. From the definition of θ_t as $\sum_{j=1}^{t-1} \lambda_j$ and the definition of \mathcal{D} we rewrite the lower bound on Δ_t as, $\Delta_t \geq -f^*(\theta_t - c\lambda'_t) + f^*(\theta_t) - c\ell_t^*(\lambda'_t)$. Since we assume that f^* is twice differentiable we can expand f^* around θ_t using Taylor expansion and get that there exists a vector θ'_t such that $\Delta_t \geq c \langle \nabla f^*(\theta_t), \lambda'_t \rangle - \frac{c^2}{2} \langle \lambda'_t, \nabla^2 f^*(\theta'_t) \lambda'_t \rangle - c\ell_t^*(\lambda'_t)$. Recall that whenever f^* is differentiable we have $\omega_t = \nabla f^*(\theta_t)$. Therefore,

$$\Delta_t \geq c (\langle \omega_t, \lambda'_t \rangle - \ell_t^*(\lambda'_t)) - \frac{c^2}{2} \langle \lambda'_t, \nabla^2 f^*(\theta'_t) \lambda'_t \rangle . \quad (10)$$

Since $\lambda'_t \in \partial_t$ and since we assume that ℓ_t is closed and convex, we can apply Lemma 1 to get that $\langle \omega_t, \lambda'_t \rangle - \ell_t^*(\lambda'_t) = \ell_t(\omega_t)$. Plugging this equality into Eq. (10) and summing over t we obtain that

$$\sum_{t=1}^T \Delta_t \geq c \sum_{t=1}^T \ell_t(\omega_t) - \frac{c^2}{2} \sum_{t=1}^T \langle \lambda'_t, \nabla^2 f^*(\theta'_t) \lambda'_t \rangle .$$

Combining the above inequality with Eq. (9) concludes our proof. \blacksquare

We now derive two types of regret bounds based on Lemma 2. Our first regret bound assumes that the average norm of λ'_t , measured with respect to the Hessian of f^* , is bounded above.

Theorem 1 *Under the same conditions of Lemma 2. Assume in addition that there exists a constant U such that $\frac{1}{T} \sum_{t=1}^T \langle \lambda'_t, \nabla^2 f^*(\theta'_t) \lambda'_t \rangle \leq 2U$. Then, for all $\omega \in \Omega$ we have,*

$$\frac{1}{T} \sum_{t=1}^T \ell_t(\omega_t) - \frac{1}{T} \sum_{t=1}^T \ell_t(\omega) \leq \frac{f(\omega)}{Tc} + cU .$$

In particular, if $c = 1/\sqrt{T}$, we obtain the bound,

$$\frac{1}{T} \sum_{t=1}^T \ell_t(\omega_t) - \frac{1}{T} \sum_{t=1}^T \ell_t(\omega) \leq \frac{f(\omega) + U}{\sqrt{T}} .$$

Proof From Lemma 2 we know that for all $\omega \in \Omega$

$$c \sum_{t=1}^T \ell_t(\omega_t) - \frac{c^2}{2} \sum_{t=1}^T \langle \lambda'_t, \nabla^2 f^*(\theta'_t) \lambda'_t \rangle \leq f(\omega) + c \sum_{t=1}^T \ell_t(\omega) .$$

Combining the above with the assumption in the theorem we get that

$$c \sum_{t=1}^T \ell_t(\omega_t) - c^2 UT \leq f(\omega) + c \sum_{t=1}^T \ell_t(\omega) .$$

Dividing the above by cT and rearranging terms concludes our proof. \blacksquare

Our second regret bound casts a different condition on the norm of sub-gradients with respect to the Hessian of f^* and is given in Appendix D.

7 Application to Online learning

In this section we demonstrate the applicability of our algorithmic framework for online learning problems. We focus on the problem of instance ranking. It can be shown that a wide range of prediction problems, such as binary classification, multiclass prediction, multilabel prediction, and label ranking, can be cast as the problem of instance ranking. In particular, at the end of this section we briefly show how to cast binary classification as ranking. In Appendix E we provide a direct description of an adaptation of our framework to the well studied problems of online binary classification and regression.

Online learning is performed in a sequence of consecutive trials. On trial t , the learner first receives an input X_t and is required to predict a target \mathbf{y}_t associated with the input. We call the pair (X_t, \mathbf{y}_t) a learning example. In instance ranking, $X_t = \{\mathbf{x}_{t,i}\}_{i=1}^{k_t}$ is a set of vectors, each of which is from an instance domain \mathcal{X} , and \mathbf{y}_t is a vector in \mathbb{R}^{k_t} . The semantic of \mathbf{y}_t is as follows. For any pair (i, j) , if $y_{t,i} > y_{t,j}$ then we say that \mathbf{y}_t ranks $\mathbf{x}_{t,i}$ ahead of $\mathbf{x}_{t,j}$. We also interpret $y_{t,i} - y_{t,j}$ as the margin confidence in which $\mathbf{x}_{t,i}$ should be ranked ahead of $\mathbf{x}_{t,j}$. For example, each $\mathbf{x}_{t,i}$ in X_t might be a representation of a movie while $y_{t,i}$ is the movie's rating, expressed as the number of stars this movie has received by a movie reviewer. As mentioned before, at each trial the learner first receives the set X_t and predicts a target vector, which we denote by $\hat{\mathbf{y}}_t \in \mathbb{R}^{k_t}$. The prediction of the learner is based on a vector ω_t , where $\hat{y}_{t,j} = \langle \omega_t, \mathbf{x}_{t,j} \rangle$. After the learner predicts the ranking $\hat{\mathbf{y}}_t$, she receives the correct ranking \mathbf{y}_t from the environment and suffers a loss according to a loss function $\ell(\omega; (X_t, \mathbf{y}_t))$. Denoting $\ell_t(\omega) = \ell(\omega; (X_t, \mathbf{y}_t))$, and assuming that ℓ_t is closed and convex, we can immediately apply our algorithmic framework from Sec. 5 and its accompanying analysis from Sec. 6 to the problem of online learning to rank.

We now describe two loss functions for ranking which generalizes the hinge-loss used in binary classification problems. Denote by E_t the set $\{(i, j) : y_{t,i} > y_{t,j}\}$. For all $(i, j) \in E_t$ we define a local hinge-loss $\ell_{i,j}(\omega; (X_t, \mathbf{y}_t)) = [(y_{t,i} - y_{t,j}) - \langle \omega, \mathbf{x}_{t,i} - \mathbf{x}_{t,j} \rangle]_+$, where $[a]_+ = \max\{a, 0\}$.

Note that $\ell_{i,j}$ is zero if ω ranks $\mathbf{x}_{t,i}$ higher than $\mathbf{x}_{t,j}$ by a sufficient confidence. Ideally, we would like $\ell_{i,j}(\omega; (X_t, \mathbf{y}_t))$ to be zero for all $(i, j) \in E_t$. If this is not the case, we are being penalized according to some combination of the pair-based losses $\ell_{i,j}$. For example, we can set $\ell(\omega; (X_t, \mathbf{y}_t))$ to be the average over the pair losses,

$$\ell^{\text{avg}}(\omega; (X_t, \mathbf{y}_t)) = \frac{1}{|E_t|} \sum_{(i,j) \in E_t} \ell_{i,j}(\omega; (X_t, \mathbf{y}_t)) . \quad (11)$$

This loss was suggested by several authors (see for example [8, 14]). Another popular approach (see for example [3]) penalizes according to the maximal loss over the individual pairs,

$$\ell^{\text{max}}(\omega; (X_t, \mathbf{y}_t)) = \max_{(i,j) \in E_t} \ell_{i,j}(\omega; (X_t, \mathbf{y}_t)) . \quad (12)$$

We can apply our algorithmic framework given in Sec. 5 for ranking, using for ℓ_t either ℓ^{avg} or ℓ^{max} . The following theorem provides us with a sufficient condition under which the regret bound from Thm. 1 holds for ranking as well.

Theorem 2 *Let f be a complexity function and assume that f^* is twice differentiable. Assume that there exists a norm $\|\cdot\|$ such that for all λ and θ we have $\langle \lambda, \nabla^2 f^*(\theta) \lambda \rangle \leq \|\lambda\|^2$. Denote by U_t the maximum over $(i, j) \in E_t$ of $\frac{1}{2} \|\mathbf{x}_{t,i} - \mathbf{x}_{t,j}\|^2$. Then, for both $\ell_t(\omega) = \ell^{\text{avg}}(\omega; (X_t, \mathbf{y}_t))$ and $\ell_t(\omega) = \ell^{\text{max}}(\omega; (X_t, \mathbf{y}_t))$, the following regret bound holds*

$$\frac{1}{T} \sum_{t=1}^T \ell_t(\omega_t) - \frac{1}{T} \sum_{t=1}^T \ell_t(\omega) \leq \frac{f(\omega) + \frac{1}{T} \sum_{t=1}^T U_t}{\sqrt{T}} .$$

If $f(\omega) = \frac{1}{2} \|\omega\|_2^2$ then the condition in the above lemma holds with the norm $\|\cdot\|_2$. If $f(\omega)$ is the relative entropy then the condition in the lemma holds with $\|\cdot\|_\infty$. We refer the reader to Appendix B for more details.

To conclude this section, we briefly show how to derive the setting of online binary classification from instance ranking. In binary classification, each example is a pair (\mathbf{x}_t, y_t) , where $\mathbf{x}_t \in \mathcal{X}$ and $y_t \in \{+1, -1\}$, and the prediction is made by $\hat{y}_t = \text{sign}(\langle \omega_t, \mathbf{x}_t \rangle)$. We can construct a set $X_t = \{\frac{1}{2} \mathbf{x}_t, -\frac{1}{2} \mathbf{x}_t\}$ and define the ranking feedback vector \mathbf{y}_t to be $\mathbf{y}_t = (1, 0)$ if $y_t = 1$ and $\mathbf{y}_t = (0, 1)$ if $y_t = -1$. It is easy to verify that both ℓ^{avg} and ℓ^{max} reduces to the well known hinge-loss function $\ell(\omega; (\mathbf{x}, y)) = [1 - y \langle \omega, \mathbf{x} \rangle]_+$. In addition, the value of U_t in Thm. 2 simply becomes $\frac{1}{2} \|\mathbf{x}_t\|^2$. For $f(\omega) = \frac{1}{2} \|\omega\|_2^2$ we obtain a new regret bound for an aggressive version of the Perceptron algorithm where the *average* squared norm of instances appearing in the bound rather than the widely used *maximal* squared norm of an instance.

8 The Game of Boosting

In this section we describe the applicability of our algorithmic framework to the analysis of boosting algorithms. A boosting algorithm uses a weak learning algorithm that generates weak-hypotheses whose performances are just slightly better than random guessing to build a strong-hypothesis which can attain an arbitrarily low error. The AdaBoost algorithm, proposed by Freund and Schapire [4], receives as input a training set of examples $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$ where for all $i \in [m]$, \mathbf{x}_i is taken from an instance domain \mathcal{X} , and y_i is a label, $y_i \in \{+1, -1\}$. The boosting proceeds in a sequence of consecutive trials. At trial t , the booster first defines a distribution, denoted ω_t , over the set of examples. Then, the booster passes the training set S along with the distribution ω_t to the weak learner. The weak learner is assumed to return a hypothesis $h_t : \mathcal{X} \rightarrow \{+1, -1\}$, such that the average error of h_t on S is slightly smaller than $\frac{1}{2}$. That is, there exists a constant $\gamma > 0$ such that,

$$\epsilon_t \stackrel{\text{def}}{=} \sum_{i=1}^m \omega_{t,i} \frac{1 - y_i h_t(\mathbf{x}_i)}{2} \leq \frac{1}{2} - \gamma . \quad (13)$$

The goal of the boosting algorithm is to invoke the weak learner several times with different distributions, and to combine the hypotheses returned by the weak learner into a final, so called strong, hypothesis whose error is small. The final hypothesis combines linearly the T hypotheses returned by the weak learner with coefficients $\alpha_1, \dots, \alpha_T$, and is defined to be the sign of $h_f(\mathbf{x})$ where $h_f(\mathbf{x}) = \sum_{t=1}^T \alpha_t h_t(\mathbf{x})$. The coefficients $\alpha_1, \dots, \alpha_T$ are determined by the booster. In AdaBoost, the initial distribution is the uniform distribution, $\omega_1 = (\frac{1}{m}, \dots, \frac{1}{m})$. At

iteration t , the value of α_t is set to be $\frac{1}{2} \log((1 - \epsilon_t)/\epsilon_t)$. The distribution is updated by the rule $\omega_{t+1,i} = \omega_{t,i} \exp(-\alpha_t y_i h_t(\mathbf{x}_i))/Z_t$, where Z_t is a normalization factor. Freund and Schapire [4] have shown that under the assumption given in Eq. (13), the error of the final strong hypothesis is at most $\exp(-2\gamma^2 T)$.

Several authors [12, 11, 6, 2] have proposed to view boosting as a coordinate-wise greedy optimization process. To do so, note first that h_f errs on an example (\mathbf{x}, y) iff $y h_f(\mathbf{x}) \leq 0$. Let $\ell_{\text{oss}}(a)$ be a monotonically non-increasing function from \mathbb{R} to \mathbb{R}_+ and assume that $\ell_{\text{oss}}(0) = 1$. Then, $\ell_{\text{oss}}(y h_f(\mathbf{x}))$ is greater than 1 whenever $y \neq \text{sign}(h_f(\mathbf{x}))$. Thus, we can restate the goal of boosting as minimizing the average loss of h_f over the training set S with respect to the variables $\alpha_1, \dots, \alpha_T$. To simplify our derivation in the sequel, we prefer to say that boosting maximizes the negation of the loss, that is,

$$\max_{\alpha_1, \dots, \alpha_T} -\frac{1}{m} \sum_{i=1}^m \ell_{\text{oss}}\left(y_i \sum_{t=1}^T \alpha_t h_t(\mathbf{x}_i)\right). \quad (14)$$

In this view, boosting is an optimization procedure which iteratively maximizes Eq. (14) with respect to the variables $\alpha_1, \dots, \alpha_T$. This view of boosting, enables the hypotheses returned by the weak learner to be general functions into the reals, $h_t : \mathcal{X} \rightarrow \mathbb{R}$ (see for instance [12]).

In this paper we view boosting as a convex repeated game between a booster and a weak learner. To motivate our construction, we would like to note that boosting algorithms define weights in two different domains: the vectors $\omega_t \in \mathbb{R}^m$ which assign weights to *examples* and the weights $\{\alpha_t : t \in [T]\}$ over *hypotheses*. In the terminology used throughout this paper, the weights $\omega_t \in \mathbb{R}^m$ are *primal* vectors while (as we show in the sequel) each weight α_t of the hypothesis h_t is related to a *dual* vector λ_t . In particular, we show that Eq. (14) is exactly the Fenchel dual of a primal problem for a convex repeated game, thus the algorithmic framework described in this paper for playing games naturally fits the problem of iteratively solving Eq. (14).

To derive the primal problem whose Fenchel dual is the problem given in Eq. (14) let us first denote by \mathbf{v}_t the vector in \mathbb{R}^m whose i th element is $v_{t,i} = y_i h_t(\mathbf{x}_i)$. For all t , we set ℓ_t to be the function $\ell_t(\omega) = [\langle \omega, \mathbf{v}_t \rangle]_+$. Intuitively, ℓ_t penalizes vectors ω which assign large weights to examples which are predicted accurately, that is $y_i h_t(\mathbf{x}_i) > 0$. In particular, if $h_t(\mathbf{x}_i) \in \{+1, -1\}$ and ω_t is a distribution over the m examples (as is the case in AdaBoost), $\ell_t(\omega_t)$ reduces to $1 - 2\epsilon_t$ (see Eq. (13)). In this case, minimizing ℓ_t is equivalent to maximizing the error of the individual hypothesis h_t over the examples. Consider the problem of minimizing $f(\omega) + c \sum_{t=1}^T \ell_t(\omega)$ and let us leave the exact form of f unspecified for now. To derive its Fenchel dual, we note that $\ell^*(-\lambda_t/c) = 0$ if there exists $\alpha_t \in [0, c]$ such that $\lambda_t = -\alpha_t \mathbf{v}_t$ and otherwise $\ell^*(-\lambda_t/c) = \infty$ (see Appendix A). Since our goal is to maximize the dual, we can restrict λ_t to take the form $\lambda_t = -\alpha_t \mathbf{v}_t$ and get that

$$\mathcal{D}(\lambda_1, \dots, \lambda_T) = -f^*\left(\sum_{t=1}^T \lambda_t\right) - 0 = -f^*\left(-\sum_{t=1}^T \alpha_t \mathbf{v}_t\right). \quad (15)$$

Now assume that $f^*(\theta)$ takes the form

$$f^*(\theta) = \Psi\left(\frac{1}{m} \sum_{i=1}^m \ell_{\text{oss}}(-\theta_i)\right) - \beta, \quad (16)$$

where $\Psi : \mathbb{R} \rightarrow \mathbb{R}$ is a monotonically increasing function and β is a scalar that ensures that $f^*(\mathbf{0}) = 0$. Then, maximizing the dual \mathcal{D} is equivalent to solving Eq. (14). In summary, we have shown that by setting $\ell_t(\omega) = [\langle \omega, \mathbf{v}_t \rangle]_+$ and setting f so that f^* is as in Eq. (16), we get that solving Eq. (14) is equivalent to maximizing the dual objective given in Eq. (15). Note that for AdaBoost, $\ell_{\text{oss}}(a) = \exp(-a)$. Thus, setting $\Psi(a) = \log(a)$ and $\beta = 0$ we obtain that $f(\omega)$ is the relative entropy between ω and the uniform distribution (see Appendix A). Minimizing the exp-loss of the strong hypothesis is therefore the dual problem of the following primal minimization problem: find a distribution over the examples, whose relative entropy to the uniform distribution is as small as possible while the correlation of the distribution with each \mathbf{v}_t is as small as possible. Since the correlation of ω with \mathbf{v}_t is negatively proportional to the error of h_t with respect to ω , we obtain that in the primal problem we are trying to *maximize* the error of each *individual* hypothesis, while in the dual problem we *minimize* the error of the *strong* hypothesis. The intuition of finding distributions which in retrospect result in large error rates of individual hypotheses was also alluded in [12, 6].

We can now apply our algorithmic framework from Sec. 5 to boosting. We describe the game with the parameters α_t and recall that in our case, $\lambda_t = -\alpha_t \mathbf{v}_t$. At the beginning of the game the booster

sets all dual variables to be zero, $\forall t \alpha_t = 0$. At trial t of the boosting game, the booster first constructs a primal weight vector $\omega_t \in \mathbb{R}^m$, which assigns importance weights to the examples in S . The primal vector ω_t is constructed as in Eq. (4), that is, $\omega_t = \nabla f^*(\theta_t)$, where $\theta_t = -\sum_i \alpha_i \mathbf{v}_i$. Then, the weak learner responds by presenting the loss function $\ell_t(\omega) = [\langle \omega, \mathbf{v}_t \rangle]_+$. Finally, the booster updates the dual variables so as to increase the dual objective function.

To analyze our game of boosting, we assume that at each trial, the hypothesis returned by the weak learner must satisfy a weak learnability assumption, particularly $\langle \omega_t, \mathbf{v}_t \rangle > 0$. Recall that the update of θ is $\theta_{t+1} = \theta_t - \alpha_t \mathbf{v}_t$. The value of α_t is found as follows. Denote $\Delta_t = \mathcal{D}(\lambda_1^{t+1}, \dots, \lambda_T^{t+1}) - \mathcal{D}(\lambda_1^t, \dots, \lambda_T^t)$. Using the definitions of \mathcal{D} and θ_t we get that $\Delta_t = f^*(\theta_t) - f^*(\theta_t - \alpha_t \mathbf{v}_t)$. As in Lemma 2, we expand f^* around θ_t using Taylor approximation and use the fact that $\omega_t = \nabla f^*(\theta_t)$ to get that there exists θ for which $\Delta_t = \alpha_t \langle \omega_t, \mathbf{v}_t \rangle - \frac{\alpha_t^2}{2} \langle \mathbf{v}_t, \nabla^2 f^*(\theta) \mathbf{v}_t \rangle$. Assume that h_t is restricted in a way such that for all θ we have $\langle \mathbf{v}_t, \nabla^2 f^*(\theta) \mathbf{v}_t \rangle \leq 1$ (see Appendix B). Then, $\Delta_t \geq \alpha_t \langle \omega_t, \mathbf{v}_t \rangle - \frac{\alpha_t^2}{2}$. The value of α_t which maximizes the above lower bound is $\alpha_t = \langle \omega_t, \mathbf{v}_t \rangle$. Thus, for this choice of α_t the dual increases by at least $\frac{1}{2}(\langle \omega_t, \mathbf{v}_t \rangle)^2$. In general, we allow α_t to be any value from the set

$$\left\{ \alpha \in \mathbb{R}_+ : f^*(\theta_t) - f^*(\theta_t - \alpha \mathbf{v}_t) \geq \frac{1}{2}(\langle \omega_t, \mathbf{v}_t \rangle)^2 \right\} . \quad (17)$$

Since the final value of the dual objective is at least $\sum_t \Delta_t$ we obtain the following corollary.

Corollary 1 *Let f be a function for which f^* takes the form given in Eq. (16), $f^*(\mathbf{0}) = 0$, and f^* is twice differentiable. Suppose that we run the game of boosting with the function f^* . For all t , denote $v_{t,i} = y_i h_t(\mathbf{x}_i)$ and assume that $\langle \omega_t, \mathbf{v}_t \rangle \geq 0$. Assume in addition that $\langle \mathbf{v}_t, \nabla^2 f^*(\theta) \mathbf{v}_t \rangle \leq 1$ for all θ and that α_t is set to be any value from the set given in Eq. (17). Then,*

$$\frac{1}{m} \sum_{i=1}^m \ell_{\text{loss}} \left(y_i \sum_{t=1}^T \alpha_t h_t(\mathbf{x}_i) \right) \leq \Psi^{-1} \left(\beta - \frac{1}{2} \sum_{t=1}^T (\langle \omega_t, \mathbf{v}_t \rangle)^2 \right) .$$

In Appendix F we derive both AdaBoost and LogitBoost from the perspective of our game of boosting and analyze them using corollary 1.

9 Related Work and Discussion

We presented a new framework for designing and analyzing algorithms for playing convex repeated games. Our framework was used for the analysis of known algorithms for both online learning and boosting settings with improved bounds. It also paves the way to new algorithms. Using duality for designing online algorithms was first suggested in [13] in the context of mistake bound analysis. Convex repeated games were termed ‘‘online convex programming’’ by Zinkevich [15]. The algorithms presented in [15] can be derived as special cases of our algorithmic framework by setting $f(\omega) = \frac{1}{2} \|\omega\|^2$. In the online learning community, there is voluminous amount of work on unified approaches for deriving online learning algorithms. We refer the reader to [7, 9, 10]. Similarly, many authors derived unifying frameworks for boosting algorithms [11, 6, 2]. Nonetheless, our general framework and the connection between game playing and Fenchel duality underscores an interesting perspective of both online learning and boosting. We believe that this viewpoint will lead to new algorithms in both domains. There are various possible extensions of the work that we did not discuss due to the lack of space. Our framework can naturally be used for the analysis of repeated games (see Appendix H and [5, 15]). The applicability of our framework to online learning can be easily extended to other prediction problems such as regression and sequence prediction. Last, we conjecture that our primal-dual view of boosting will lead to new methods for regularizing boosting algorithms, thus improving their generalization capabilities.

References

- [1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [2] M. Collins, R.E. Schapire, and Y. Singer. Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, 47(2/3):253–285, 2002.
- [3] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive aggressive algorithms. *Journal of Machine Learning Research*, 7, Mar 2006.

- [4] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory: Second European Conference, EuroCOLT '95*, pages 23–37. Springer-Verlag, 1995.
- [5] Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, pages 325–332, 1996.
- [6] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–374, April 2000.
- [7] A. J. Grove, N. Littlestone, and D. Schuurmans. General convergence results for linear discriminant updates. *Machine Learning*, 43(3):173–210, 2001.
- [8] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2002.
- [9] J. Kivinen, A. J. Smola, and R. C. Williamson. Online learning with kernels. *IEEE Transactions on Signal Processing*, 52(8):2165–2176, 2002.
- [10] J. Kivinen and M. Warmuth. Relative loss bounds for multidimensional regression problems. *Journal of Machine Learning*, 45(3):301–329, July 2001.
- [11] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. Functional gradient techniques for combining hypotheses. In *Advances in Large Margin Classifiers*. MIT Press, 1999.
- [12] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):1–40, 1999.
- [13] S. Shalev-Shwartz and Y. Singer. Online learning meets optimization in the dual. In *Proceedings of the Nineteenth Annual Conference on Computational Learning Theory*, 2006.
- [14] J. Weston and C. Watkins. Support vector machines for multi-class pattern recognition. In *Proceedings of the Seventh European Symposium on Artificial Neural Networks*, April 1999.
- [15] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*, 2003.

Appendices for the paper Convex Repeated Games and Fenchel Duality

A Some Fenchel conjugate pairs

In this section we list a few useful Fenchel-conjugate pairs.

Half-squared-norm Let $\|\cdot\|$ be any norm on \mathbb{R}^n and let $f(\omega) = \frac{1}{2}\|\omega\|^2$ with $\Omega = \mathbb{R}^n$. Then $f^*(\theta) = \frac{1}{2}\|\theta\|_*^2$ where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$. The domain of f^* is also \mathbb{R}^n . For example, if $f(\omega) = \frac{1}{2}\|\omega\|_2^2$ then $f^*(\theta) = \frac{1}{2}\|\theta\|_2^2$ since the ℓ_2 norm is dual to itself. For a proof see pp. 93-94 in [1].

Hinge-loss Let $f(\omega) = [\gamma - \langle \omega, \mathbf{x} \rangle]_+$ where $\gamma \in \mathbb{R}_+$ and $\mathbf{x} \in \mathbb{R}^n$ with $\Omega = \mathbb{R}^n$. Then,

$$f^*(\theta) = \begin{cases} -\gamma\alpha & \text{if } \theta \in \{-\alpha\mathbf{x} : \alpha \in [0, 1]\} \\ \infty & \text{otherwise} \end{cases}$$

To show the above, recall that

$$f^*(\theta) = \sup_{\omega \in \mathbb{R}^n} \langle \omega, \theta \rangle - [\gamma - \langle \omega, \mathbf{x} \rangle]_+ . \quad (18)$$

Consider two cases. Case I: $\theta = -\alpha\mathbf{x}$ for some $\alpha \in \mathbb{R}$. First note that in this case the objective to maximize in Eq. (18) becomes

$$-\alpha \langle \omega, \mathbf{x} \rangle - [\gamma - \langle \omega, \mathbf{x} \rangle]_+ = \begin{cases} -\alpha \langle \omega, \mathbf{x} \rangle & \langle \omega, \mathbf{x} \rangle \geq \gamma \\ (1-\alpha)\langle \omega, \mathbf{x} \rangle - \gamma & \langle \omega, \mathbf{x} \rangle \leq \gamma \end{cases} .$$

Denote $a = \langle \omega, \mathbf{x} \rangle$ and note that a can take any value in \mathbb{R} . Thus, we obtain that

$$f^*(\theta) = \max \left\{ \sup_{a \leq \gamma} ((1-\alpha)a - \gamma), \sup_{a \geq \gamma} (-\alpha a) \right\} = \begin{cases} -\gamma\alpha & \alpha \in [0, 1] \\ \infty & \alpha \notin [0, 1] \end{cases} .$$

We now turn to case II in which there does not exist $\alpha \in \mathbb{R}$ such that $\theta = -\alpha\mathbf{x}$. In this case, we can rewrite θ as $\theta = \frac{\langle \theta, \mathbf{x} \rangle}{\|\mathbf{x}\|^2} \mathbf{x} + \mathbf{v}$ where $\mathbf{v} \in \{\omega : \langle \omega, \mathbf{x} \rangle = 0\}$ and \mathbf{v} must not equal to zero. Thus, setting $\omega = a\mathbf{v}$ in the objective in Eq. (18) gives

$$a \langle \mathbf{v}, \theta \rangle - [\gamma - a \langle \mathbf{v}, \mathbf{x} \rangle]_+ = a \|\mathbf{v}\|^2 - \gamma ,$$

which also tends to ∞ when $a \rightarrow \infty$.

Relative Entropy and log-sum-exp Let $\Omega = \{\omega \in \mathbb{R}_+^n : \sum_{i=1}^n \omega_i = 1\}$ and let

$$f(\omega) = \sum_{i=1}^n \omega_i \log \left(\frac{\omega_i}{1/n} \right) .$$

Then,

$$f^*(\theta) = \log \left(\frac{1}{n} \sum_{i=1}^n \exp(\theta_i) \right) . \quad (19)$$

For a proof see p. 93 in [1].

binary-entropy and the log-loss Let $\Omega = \{\omega \in \mathbb{R}^n : \forall i \in [n], \omega_i \in [0, 1]\}$ and let

$$f(\omega) = - \sum_{i=1}^n (\omega_i \log(\omega_i) + (1 - \omega_i) \log(1 - \omega_i)) .$$

Then,

$$f^*(\theta) = \sum_{i=1}^n \log(1 + e^{\theta_i}) . \quad (20)$$

To show this we can use the fact that for differentiable functions we have that $\nabla f(\omega)$ is the inverse of the function $\nabla f^*(\theta)$, which can be easily verified for the above conjugate pair.

The effect of scaling and shifting We conclude this section with the following useful property of conjugate pairs. Let f be a function and let f^* be its Fenchel conjugate. For $a > 0$ and $b \in \mathbb{R}$, the Fenchel conjugate of $g(\omega) = af(\omega) + b$ is $g^*(\theta) = af^*(\omega/a) - b$. For a proof, see page 95 in [1].

B Bounding $\langle \boldsymbol{\lambda}, \nabla^2 f^*(\boldsymbol{\theta}) \boldsymbol{\lambda} \rangle$

In this section we provide bounds on the quadratic pattern $\langle \boldsymbol{\lambda}, \nabla^2 f^*(\boldsymbol{\theta}) \boldsymbol{\lambda} \rangle$ for several conjugate functions $f^*(\boldsymbol{\theta})$. First, if $f^*(\boldsymbol{\theta}) = \frac{1}{2} \|\boldsymbol{\theta}\|_2^2$ then clearly $\langle \boldsymbol{\lambda}, \nabla^2 f^*(\boldsymbol{\theta}) \boldsymbol{\lambda} \rangle = \|\boldsymbol{\lambda}\|_2^2$. Note that in this case $f(\boldsymbol{\theta})$ can be rewritten as

$$f^*(\boldsymbol{\theta}) = \Psi \left(\sum_{i=1}^n \phi(\theta_i) \right), \quad (21)$$

where $\Psi(a) = \frac{1}{2}a$ and $\phi(a) = a^2$. The following lemma provides us with a general tool for bounding $\langle \boldsymbol{\lambda}, \nabla^2 f^*(\boldsymbol{\theta}) \boldsymbol{\lambda} \rangle$ for functions of the form given in Eq. (21).

Lemma 3 *Assume that f^* can be written as in Eq. (21), where ϕ and Ψ are twice differentiable scalar functions. Denote by $\phi', \phi'', \Psi', \Psi''$ the first and second order derivatives of Ψ and ϕ . If $\Psi''(\sum_i \phi(\theta_i)) \leq 0$ for all $\boldsymbol{\theta}$ then,*

$$\langle \boldsymbol{\lambda}, \nabla^2 f^*(\boldsymbol{\theta}) \boldsymbol{\lambda} \rangle \leq \Psi' \left(\sum_{r=1}^n \phi(\theta_r) \right) \sum_{i=1}^n \phi''(\theta_i) \lambda_i^2.$$

Proof Denote $H = \nabla^2 f^*(\boldsymbol{\theta})$. Using the chain rule we get that,

$$\nabla_i f^*(\boldsymbol{\theta}) = \Psi' \left(\sum_{r=1}^n \phi(\theta_r) \right) \phi'(\theta_i).$$

Therefore, the value of the element (i, j) of the Hessian for $i \neq j$ is,

$$H_{i,j}(\boldsymbol{\theta}) = \Psi'' \left(\sum_{r=1}^n \phi(\theta_r) \right) \phi'(\theta_i) \phi'(\theta_j),$$

and the i 'th diagonal element of the Hessian is,

$$H_{i,i}(\boldsymbol{\theta}) = \Psi'' \left(\sum_{r=1}^n \phi(\theta_r) \right) (\phi'(\theta_i))^2 + \Psi' \left(\sum_{r=1}^n \phi(\theta_r) \right) \phi''(\theta_i).$$

We therefore get that,

$$\begin{aligned} \langle \boldsymbol{\lambda}, H(\boldsymbol{\theta}) \boldsymbol{\lambda} \rangle &= \Psi'' \left(\sum_{r=1}^n \phi(\theta_r) \right) \left(\sum_i \phi'(\theta_i) \lambda_i \right)^2 + \Psi' \left(\sum_{r=1}^n \phi(\theta_r) \right) \sum_i \phi''(\theta_i) \lambda_i^2 \\ &\leq \Psi' \left(\sum_{r=1}^n \phi(\theta_r) \right) \sum_i \phi''(\theta_i) \lambda_i^2, \end{aligned}$$

where the last inequality follows from the assumption that $\Psi''(\sum_r \phi(\theta_r)) \leq 0$. ■

Note that if we apply Lemma 3 to the function $\frac{1}{2} \|\boldsymbol{\theta}\|_2^2$ we indeed get that $\langle \boldsymbol{\lambda}, \nabla^2 f^*(\boldsymbol{\theta}) \boldsymbol{\lambda} \rangle \leq \|\boldsymbol{\lambda}\|_2^2$. We now consider the log-sum-exp function given in Eq. (19). Applying Lemma 3 with $\Psi(a) = \log(a)$ and $\phi(a) = \exp(a)$ (and note that $\Psi''(\sum_i \phi(\theta_i)) \leq 0$) gives that,

$$\langle \boldsymbol{\lambda}, \nabla^2 f^*(\boldsymbol{\theta}) \boldsymbol{\lambda} \rangle \leq \sum_{i=1}^n \frac{\exp(\theta_i)}{\sum_{r=1}^n \exp(\theta_r)} \lambda_i^2 \leq \max_{i \in [n]} \lambda_i^2 = \|\boldsymbol{\lambda}\|_\infty^2.$$

Next, we consider the log-loss function given in Eq. (20). We can rewrite the log-loss function as in Eq. (21) with $\Psi(a) = a$ and $\phi(a) = \log(1 + e^{-a})$. The conditions in Lemma 3 trivially hold since $\Psi''(a) = 0$ for all a . In addition,

$$\phi'(a) = \frac{e^a}{1 + e^a}, \quad \text{and} \quad \phi''(a) = \frac{e^a}{1 + e^a} \left(1 - \frac{e^a}{1 + e^a} \right) \leq \frac{1}{4}.$$

Therefore, Lemma 3 gives that for the log-loss,

$$\langle \boldsymbol{\lambda}, \nabla^2 f^*(\boldsymbol{\theta}) \boldsymbol{\lambda} \rangle \leq \frac{1}{4} \sum_{i=1}^n \lambda_i^2.$$

C Technical Proofs

Proof of Lemma 1 Since $\lambda' \in \partial f(\omega')$, we know that $f(\omega) - f(\omega') \geq \langle \lambda', \omega - \omega' \rangle$ for all $\omega \in \Omega$. Equivalently

$$\langle \lambda', \omega' \rangle - f(\omega') \geq \sup_{\omega \in \Omega} (\langle \lambda', \omega \rangle - f(\omega)) .$$

The right-hand side of the above equals to $f^*(\lambda')$ and thus,

$$\langle \lambda', \omega' \rangle - f(\omega') \geq f^*(\lambda') \quad \Rightarrow \quad \langle \lambda', \omega' \rangle - f^*(\lambda') \geq f(\omega') . \quad (22)$$

The assumption that f is closed and convex implies that f is the Fenchel conjugate of f^* . Thus,

$$f(\omega') = \sup_{\lambda} (\langle \lambda, \omega' \rangle - f^*(\lambda)) \geq \langle \lambda', \omega' \rangle - f^*(\lambda') .$$

Combining the above with Eq. (22) gives,

$$\langle \lambda', \omega' \rangle - f^*(\lambda') \geq f(\omega') \quad \text{and} \quad f(\omega') \geq \langle \lambda', \omega' \rangle - f^*(\lambda') .$$

Therefore, each of the two inequalities above must hold with equality which concludes the proof. ■

Proof of Thm. 2 To apply Thm. 1 it suffices to show that for all t there exists a sub-gradient $\lambda'_t \in \partial_t$ such that for all θ we have $\langle \lambda'_t, \nabla^2 f^*(\theta) \lambda'_t \rangle \leq 2U_t$. We start with the case $\ell_t(\omega) = \ell^{\text{avg}}(\omega; (X_t, \mathbf{y}_t))$. For all $(i, j) \in E_t$, let $\lambda_{i,j} \in \partial \ell_{i,j}(\omega_t; (X_t, \mathbf{y}_t))$. We now show that $\lambda'_t = \frac{1}{|E_t|} \sum_{(i,j)} \lambda_{i,j}$ is a sub-gradient of ℓ_t at ω_t . To do so, note that for all $\omega \in \Omega$,

$$\begin{aligned} \ell_t(\omega) - \ell_t(\omega_t) &= \frac{1}{|E_t|} \sum_{(i,j) \in E_t} (\ell_{i,j}(\omega; (X_t, \mathbf{y}_t)) - \ell_{i,j}(\omega_t; (X_t, \mathbf{y}_t))) \\ &\geq \frac{1}{|E_t|} \sum_{(i,j) \in E_t} \langle \lambda_{i,j}, \omega - \omega_t \rangle = \langle \lambda'_t, \omega - \omega_t \rangle . \end{aligned}$$

Thus, λ'_t is indeed a sub-gradient of ℓ_t at ω_t . Next, we show that $\langle \lambda'_t, \nabla^2 f^*(\theta) \lambda'_t \rangle \leq 2U_t$. First we use Cauchy-Schwartz inequality to get

$$\|\lambda'_t\| \leq \frac{1}{|E_t|} \sum_{(i,j) \in E_t} \|\lambda_{i,j}\| \leq \max_{(i,j) \in E_t} \|\lambda_{i,j}\| . \quad (23)$$

In addition, the definition of $\ell_{i,j}$ implies that $\lambda_{i,j}$ is either the zero vector or the vector $(\mathbf{x}_{t,i} - \mathbf{x}_{t,j})$. Therefore, $\|\lambda_{i,j}\| \leq \|\mathbf{x}_{t,i} - \mathbf{x}_{t,j}\|$. Combining the last inequality with Eq. (23) and using the definition of U_t we get that $\|\lambda'_t\| \leq \sqrt{2U_t}$. Using the assumption in the theorem we conclude that

$$\langle \lambda'_t, \nabla^2 f^*(\theta) \lambda'_t \rangle \leq \|\lambda'_t\|^2 \leq 2U_t .$$

Thus, the condition in Thm. 1 holds with $U = \frac{1}{T} \sum_t U_t$ and the regret bound in the theorem follows directly. This concludes the proof for the case $\ell_t(\omega) = \ell^{\text{avg}}(\omega; (X_t, \mathbf{y}_t))$. Turning to the case $\ell_t(\omega) = \ell^{\text{max}}(\omega; (X_t, \mathbf{y}_t))$, we note that for all t , there exists $(i, j) \in E_t$ such that $\ell_t(\omega_t) = \ell_{i,j}(\omega_t; (X_t, \mathbf{y}_t))$. Let λ'_t be a sub-gradient of $\ell_{i,j}$ at ω_t . We show that λ'_t is also a sub-gradient of ℓ_t at ω_t . This follows directly from the fact that $\ell_t(\omega_t) = \ell_{i,j}(\omega_t; (X_t, \mathbf{y}_t))$ and that for all other ω we have $\ell_t(\omega) \geq \ell_{i,j}(\omega; (X_t, \mathbf{y}_t))$. Thus, for all ω ,

$$\ell_t(\omega) - \ell_t(\omega_t) \geq \ell_{i,j}(\omega; (X_t, \mathbf{y}_t)) - \ell_{i,j}(\omega_t; (X_t, \mathbf{y}_t)) \geq \langle \lambda'_t, \omega - \omega_t \rangle ,$$

where the last inequality follows from the fact that λ'_t is a sub-gradient of $\ell_{i,j}$ at ω_t . In addition, the assumptions in the theorem imply that $\langle \lambda'_t, \nabla^2 f^*(\theta) \lambda'_t \rangle \leq 2U_t$. Thus, the condition in Thm. 1 holds for this case as well and our proof is completed. ■

D A second Regret Bound

Theorem 3 *Under the same conditions of Lemma 2. Assume in addition that there exists a constant U such that for all t*

$$\langle \boldsymbol{\lambda}'_t, \nabla^2 f^*(\boldsymbol{\theta}'_t) \boldsymbol{\lambda}'_t \rangle \leq 2U \ell_t(\boldsymbol{\omega}_t) .$$

Let $\epsilon = cU/(1 - cU)$. Then, for all $\boldsymbol{\omega} \in \Omega$ we have,

$$\frac{1}{T} \sum_{t=1}^T \ell_t(\boldsymbol{\omega}_t) - \frac{1}{T} \sum_{t=1}^T \ell_t(\boldsymbol{\omega}) \leq \frac{U(1+\epsilon)^2}{T\epsilon} f(\boldsymbol{\omega}) + \epsilon \frac{1}{T} \sum_{t=1}^T \ell_t(\boldsymbol{\omega}) .$$

In particular, for $\epsilon = 1/\sqrt{T}$ we obtain that

$$\frac{1}{T} \sum_{t=1}^T \ell_t(\boldsymbol{\omega}_t) - \frac{1}{T} \sum_{t=1}^T \ell_t(\boldsymbol{\omega}) \leq \frac{U \left(1 + \frac{1}{\sqrt{T}}\right)^2 f(\boldsymbol{\omega}) + \frac{1}{T} \sum_{t=1}^T \ell_t(\boldsymbol{\omega})}{\sqrt{T}} .$$

Proof Combining the assumption in the theorem with Lemma 2 we get that for all $\boldsymbol{\omega} \in \Omega$

$$c(1 - cU) \sum_{t=1}^T \ell_t(\boldsymbol{\omega}_t) \leq f(\boldsymbol{\omega}) + c \sum_{t=1}^T \ell_t(\boldsymbol{\omega}) .$$

Rearranging terms we get that

$$\frac{1}{T} \sum_{t=1}^T \ell_t(\boldsymbol{\omega}_t) \leq \frac{f(\boldsymbol{\omega})}{Tc(1 - cU)} + \frac{1}{1 - cU} \frac{1}{T} \sum_{t=1}^T \ell_t(\boldsymbol{\omega}) . \quad (24)$$

From the definition of ϵ we know that

$$1 + \epsilon = 1 + \frac{cU}{1 - cU} = \frac{1 - cU + cU}{1 - cU} = \frac{1}{1 - cU} , \quad (25)$$

and that

$$\frac{U(1 + \epsilon)^2}{\epsilon} = \frac{U}{cU(1 - cU)} = \frac{1}{c(1 - cU)} . \quad (26)$$

Plugging Eq. (25) and Eq. (26) into Eq. (24) we obtain

$$\frac{1}{T} \sum_{t=1}^T \ell_t(\boldsymbol{\omega}_t) \leq \frac{U(1 + \epsilon)^2 f(\boldsymbol{\omega})}{T\epsilon} + (1 + \epsilon) \frac{1}{T} \sum_{t=1}^T \ell_t(\boldsymbol{\omega}) ,$$

which concludes our proof. ■

E Simple Online Prediction Problems

E.1 Online Binary Classification with the Hinge-Loss

In binary classification, the target set is $\mathcal{Y} = \{+1, -1\}$. Usually, the prediction of the learner is defined to be $\text{sign}(\langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle)$. A popular choice of a loss function is the hinge-loss defined as $\ell_t(\boldsymbol{\omega}) = [1 - y_t \langle \boldsymbol{\omega}, \mathbf{x}_t \rangle]_+$, where $[a]_+ = \max\{0, a\}$. Note that whenever $\ell_t(\boldsymbol{\omega}_t) > 0$ we have $-y_t \mathbf{x}_t \in \partial_t$ and that if $\ell_t(\boldsymbol{\omega}_t) = 0$ then $\mathbf{0} \in \partial_t$. Consider the dual update rule given in Eq. (7). If we set $f(\boldsymbol{\omega}) = \frac{1}{2} \|\boldsymbol{\omega}\|_2^2$ then $f^*(\boldsymbol{\theta}) = \frac{1}{2} \|\boldsymbol{\theta}\|_2^2$ (see Appendix A). Therefore, the update in Eq. (7) of the dual variables is equivalent to the following update of the primal variables: $\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t + c y_t \mathbf{x}_t$ whenever $\ell_t(\boldsymbol{\omega}_t) > 0$ and otherwise $\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t$. The resulting algorithm is an aggressive version of the Perceptron algorithm for binary classification. If we apply the update in Eq. (8) we obtain the PA-I algorithm for binary classification described in [3].

Turning to the analysis of the aggressive Perceptron (and of the PA-I algorithm) we note that for all $\boldsymbol{\theta}$ the matrix $\nabla^2 f^*(\boldsymbol{\theta})$ is the identity matrix. Therefore, for all $\boldsymbol{\lambda} \in \partial_t$ we have that $\langle \boldsymbol{\lambda}, \nabla^2 f^*(\boldsymbol{\theta}) \boldsymbol{\lambda} \rangle =$

$\|\lambda\|_2^2 \leq \|\mathbf{x}_t\|_2^2$. We can now apply Thm. 1 and get the following regret bound for the aggressive Perceptron

$$\forall \omega \in \Omega, \quad \frac{1}{T} \sum_{t=1}^T \ell_t(\omega_t) - \frac{1}{T} \sum_{t=1}^T \ell_t(\omega) \leq \frac{\|\omega\|_2^2 + \frac{1}{T} \sum_{t=1}^T \|\mathbf{x}_t\|_2^2}{2\sqrt{T}}. \quad (27)$$

Previous bounds for the Perceptron compared the number of prediction *mistakes* made by the Perceptron to the cumulative *loss* of the fixed competitor ω . In contrast, our regret bound compares the hinge-loss of the aggressive Perceptron to the hinge-loss of the fixed competitor. In addition, previous bounds for the Perceptron involves the maximal squared norm of an instance. In contrast, our bound is based on the average squared norm of the instances which may be significantly smaller than the maximal squared norm.

Finally, we would like to note in passing that choosing other complexity functions result in other aggressive variants of well known algorithms. For example, if Ω is the probability simplex and $f(\omega)$ is the relative entropy between ω and the uniform distribution $(\frac{1}{n}, \dots, \frac{1}{n})$ then repeating the above derivation result in an aggressive variant of the EG/Winnow algorithm (see [10] and the references therein). It can be shown that applying Thm. 1 to this algorithm gives the regret bound

$$\forall \omega \in \Omega, \quad \frac{1}{T} \sum_{t=1}^T \ell_t(\omega_t) - \frac{1}{T} \sum_{t=1}^T \ell_t(\omega) \leq \frac{2 \log(n) + \frac{1}{T} \sum_{t=1}^T \|\mathbf{x}_t\|_\infty^2}{2\sqrt{T}}. \quad (28)$$

E.2 Online Regression with the ε -insensitive loss and with the squared loss

In online regression, the target set is the reals, $\mathcal{Y} = \mathbb{R}$, and the prediction of the algorithm is simply $\langle \omega_t, \mathbf{x}_t \rangle$. A popular loss function for regression is the absolute loss defined by $\ell(\omega; (\mathbf{x}_t, y_t)) = |\langle \omega, \mathbf{x}_t \rangle - y_t|$. A variant of this loss does not penalize the learner for small discrepancies and is defined as $\ell(\omega; (\mathbf{x}_t, y_t); \varepsilon) = [|\langle \omega, \mathbf{x}_t \rangle - y_t| - \varepsilon]_+$. As in binary classification, we can set $\ell_t(\omega) = \ell(\omega; (\mathbf{x}_t, y_t); \varepsilon)$ and apply our algorithmic framework from Sec. 5 along with its accompanying analysis from Sec. 6. In particular, it is easily verified that the bound in Eq. (27) holds for regression as well.

Another popular loss function for regression is the squared loss defined as $\ell(\omega; (\mathbf{x}_t, y_t)) = \frac{1}{2} (\langle \omega, \mathbf{x}_t \rangle - y_t)^2$. This loss function is differentiable and we have that $\nabla \ell_t(\omega_t) = (\langle \omega_t, \mathbf{x}_t \rangle - y_t) \mathbf{x}_t$. Thus, the condition $\|\mathbf{x}_t\|_2^2 \leq U$ implies that the condition given in Thm. 3 holds and we obtain the regret bound

$$\frac{1}{T} \sum_{t=1}^T \ell_t(\omega_t) - \frac{1}{T} \sum_{t=1}^T \ell_t(\omega) \leq \frac{U(1+\epsilon)^2}{m\epsilon} f(\omega) + \epsilon \frac{1}{T} \sum_{t=1}^T \ell_t(\omega),$$

where $\epsilon = cU(1 - cU)$.

F AdaBoost and LogitBoost

In this section we derive both the AdaBoost and the LogitBoost algorithms from our game of boosting framework. By setting $\ell_{\text{oss}}(a) = \exp(-a)$, $\Psi(a) = \log(a)$, and $\beta = 0$, we get that f^* is twice differentiable and that $f^*(\mathbf{0}) = 0$. From the definition of ω_t given in Eq. (4) we get that

$$\omega_{t,i} = \frac{\exp(\theta_{t,i})}{\sum_{j=1}^m \exp(\theta_{t,j})}.$$

Note that for $t = 1$ we have $\theta_t = \mathbf{0}$ and thus ω_1 is the uniform distribution over the m examples. In addition, in Appendix B we show that if $h_t(\mathbf{x}_i) \in [+1, -1]$ for all i then $\langle \mathbf{v}_t, \nabla^2 f^*(\theta) \mathbf{v}_t \rangle \leq 1$ for all θ . Suppose that we update α_t to be the maximizer of the increase in the dual, $f^*(\theta_t - \alpha \mathbf{v}_t) - f^*(\theta_t)$, over α . This value of α_t is clearly in the set given in Eq. (17). Using the definition of ω_t we have that

$$\alpha_t = \underset{\alpha \geq 0}{\operatorname{argmin}} f^*(\theta_t - \alpha \mathbf{v}_t) - f^*(\theta_t) = \underset{\alpha \geq 0}{\operatorname{argmin}} \sum_{i=1}^m \omega_{t,i} e^{-\alpha_i y_i h_t(\mathbf{x}_i)}.$$

Schapire and Singer [12] have shown that whenever $h_t(\mathbf{x}_i) \in \{+1, -1\}$ the solution of the above equation is $\alpha_t = \frac{1}{2} \log((1 - \epsilon_t)/\epsilon_t)$ where ϵ_t is as defined in Eq. (13). Furthermore, the update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_t \mathbf{v}_t$ implies that $\omega_{t+1,i} = \omega_{t,i} e^{-\alpha_t y_i h_t(\mathbf{x}_i)} / Z_t$ where Z_t is a normalization factor. We have thus derived the AdaBoost algorithm from our game of boosting. Moreover, as we have shown before, for $h_t(\mathbf{x}_i) \in \{+1, -1\}$, the weak learnability assumption given in Eq. (13) implies that, $\langle \boldsymbol{\omega}_t, \mathbf{v}_t \rangle = 1 - 2\epsilon_t \geq 2\gamma$. Thus, corollary 1 gives that

$$\frac{1}{m} \sum_{i=1}^m \exp(-y_i h_f(\mathbf{x}_i)) \leq \exp(-2\gamma^2 T) ,$$

which is identical to the original bound for AdaBoost. The LogitBoost algorithm can be derived from our game of boosting in a similar manner by setting $\Psi(a) = a$, $\ell_{\text{oss}}(y_i H(\mathbf{x}_i)) = \log(1 + \exp(-y_i H(\mathbf{x}_i)))$, and $\beta = \log(2)$. At each step of LogitBoost, the weight are set to be

$$\omega_{t,i} = \frac{1}{m(1 + e^{-\theta_{t,i}})} .$$

Thus, initially the weights are $(\frac{1}{2m}, \dots, \frac{1}{2m})$. It can be shown (see again Appendix B) that $\langle \mathbf{v}_t, \nabla^2 f^*(\boldsymbol{\theta}) \mathbf{v}_t \rangle \leq \frac{1}{4m} \sum_i \mathbf{v}_{t,i}^2$. Thus, if the average value of $h_t(\mathbf{x}_i)^2$ is smaller than 4, we can apply corollary 1 and get that

$$\frac{1}{m} \sum_{i=1}^m \log(1 + e^{-y_i h_f(\mathbf{x}_i)}) \leq \log(2) - \frac{1}{2} \sum_{t=1}^T (\langle \boldsymbol{\omega}_t, \mathbf{v}_t \rangle)^2 .$$

Therefore, the number of iterations, T , on which $\langle \boldsymbol{\omega}_t, \mathbf{v}_t \rangle \geq 2\gamma$ can not exceed $\log(2)/(4\gamma^2)$.

G Boosting with regularization

The objective function we maximizes in Boosting is as in Eq. (15). In many situations, we would like to avoid over-fitting effects by adding a regularization term to the objective. One way is simply to force α_t to be at most c (thus enforcing $\|\boldsymbol{\alpha}\|_\infty$ to be at most c). Deriving this boosting method follows directly from our analysis before. Another possibility is to introduce the objective

$$\mathcal{D}(\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_T) = -f^* \left(-\sum_{t=1}^T \alpha_t \mathbf{v}_t \right) - \mu \sum_{t=1}^T \alpha_t , \quad (29)$$

where μ is a trade-off parameter. Here, we penalize $\boldsymbol{\alpha}$ with large ℓ_1 norm.

We now show how a simple change of the definition of ℓ_t yields the dual problem given in Eq. (29). Let us redefine ℓ_t to be

$$\ell_t(\boldsymbol{\omega}) = [\langle \boldsymbol{\omega}, \mathbf{v}_t \rangle - \mu]_+ . \quad (30)$$

Again, based on Appendix A we know that $\ell^*(-\boldsymbol{\lambda}_t/c)$ is $\mu \alpha_t$ if there exists $\alpha_t \in [0, c]$ such that $\boldsymbol{\lambda}_t = -\alpha_t \mathbf{v}_t$ and otherwise, $\ell^*(-\boldsymbol{\lambda}_t/c) = \infty$. Therefore, the dual of the primal problem of minimizing $f(\boldsymbol{\omega}) + c \sum_t \ell_t(\boldsymbol{\omega})$, where ℓ_t is as in Eq. (30) is to maximize Eq. (29).

Recall that whenever $h_t(\mathbf{x}_i) \in \{+1, -1\}$ we have that $\langle \boldsymbol{\omega}_t, \mathbf{v}_t \rangle = 1 - 2\epsilon_t$. Thus, the value of $\ell_t(\boldsymbol{\omega}_t) = [\langle \boldsymbol{\omega}_t, \mathbf{v}_t \rangle - \mu]_+ = [(1 - \mu) - 2\epsilon_t]_+$. That is, $\ell_t(\boldsymbol{\omega}_t) > 0$ only if $\epsilon_t \leq \frac{1}{2} - \frac{\mu}{2}$, so we require a stronger weak learnability criterion (we show in the sequel that to obtain progress we must have positive value for $\ell_t(\boldsymbol{\omega}_t)$).

To calculate α_t , we can repeat the ideas in the paper to see that if $\langle \mathbf{v}_t, \nabla^2 f^*(\boldsymbol{\theta}) \mathbf{v}_t \rangle \leq 1$ then

$$\Delta_t \geq \alpha_t \ell_t(\boldsymbol{\omega}_t) - \frac{\alpha_t^2}{2} .$$

Thus, if $\alpha_t = \ell_t(\boldsymbol{\omega}_t)$ (assuming that $c \gg 1$) we obtain the minimal progress $\Delta_t \geq \frac{1}{2} (\ell_t(\boldsymbol{\omega}_t))^2$. Similar to Corollary 1 we can obtain the following bound

$$\frac{1}{m} \sum_{i=1}^m \ell_{\text{oss}} \left(y_i \sum_{t=1}^T \alpha_t h_t(\mathbf{x}_i) \right) \leq \Psi^{-1} \left(\beta - \frac{1}{2} \sum_{t=1}^T (\ell_t(\boldsymbol{\omega}_t))^2 - \mu \sum_{t=1}^T \alpha_t \right) .$$

For the exp-loss and if we set $\alpha_t = \ell_t$ we obtain

$$\frac{1}{m} \sum_{i=1}^m \ell_{\text{oss}} \left(y_i \sum_{t=1}^T \alpha_t h_t(\mathbf{x}_i) \right) \leq \exp \left(-\frac{1}{2} \sum_{t=1}^T \ell_t(\boldsymbol{\omega}_t) (\ell_t(\boldsymbol{\omega}_t) + 2\mu) \right) .$$

Note that we do not make any progress if $\ell_t(\boldsymbol{\omega}_t) = 0$, which is equivalent (for $h_t \in \{+1, -1\}$) to the condition $\epsilon_t \geq \frac{1}{2} - \frac{\mu}{2}$. Thus, adding the regularization enforce stronger evidence for adding new hypotheses and as a result, the convergence rate is slower.

H Application to Game Theory

In this section we underscore the applicability of our framework for game theory. Specifically, we show that our framework can be used for playing repeated games with mixed strategies. A repeated game is performed in a sequence of consecutive trials. At trial t , the first player chooses an action $a \in A$, where A is a finite set of predefined actions. Then, the second player chooses an action $b \in B$, where B is another finite set of actions. After each trial, the first player suffers a loss according to a function $\ell_{\text{oss}} : A \times B \rightarrow \mathbb{R}$. Since we assume that A and B are finite sets, we can simply assume that $A = [m]$ and $B = [n]$ and that $\ell_{\text{oss}}(a, b) = M_{a,b}$ for some fixed (but unknown) matrix $M \in \mathbb{R}^{m,n}$.

As mentioned before, each player chooses a *single* action at trial t . Choosing individual actions by a player is often called playing with *pure* strategies. Usually, we allow the players to randomized their choice of action. This is done by allowing the players to define a distribution over the set of actions rather than to choose a single action. That is, at trial t the first player chooses a distribution $\boldsymbol{\omega}_t$ over A , where $\boldsymbol{\omega}_t$ is a vector in the m -dimensional simplex. Then, the second player chooses a distribution \mathbf{x}_t over B , where \mathbf{x}_t is a vector in the n -dimensional simplex. The expected loss of the first player on this trial is defined to be

$$\langle \boldsymbol{\omega}_t, M\mathbf{x}_t \rangle = \sum_{i=1}^m \sum_{j=1}^n \omega_{t,i} x_{t,j} M_{i,j} .$$

Repeated games in which players are allowed to define distributions over actions are called plays with *mixed* strategies.

We now show how our template algorithm for playing convex repeated games can be utilized for playing a repeated game with mixed strategies. Set $\Omega = \{\boldsymbol{\omega} \in \mathbb{R}^m : \omega_i \geq 0, \sum_i \omega_i = 1\}$. Let $f : \Omega \rightarrow \mathbb{R}$ be a function such that $\inf_{\boldsymbol{\omega} \in \Omega} f(\boldsymbol{\omega}) = 0$. For example, f can be the relative entropy between $\boldsymbol{\omega}$ and the uniform distribution $(\frac{1}{m}, \dots, \frac{1}{m})$

$$f(\boldsymbol{\omega}) = \sum_{i=1}^m \omega_i \log \left(\frac{\omega_i}{1/m} \right) . \quad (31)$$

The conjugate of f is (see Appendix A)

$$f^*(\boldsymbol{\theta}) = \log \left(\frac{1}{m} \sum_{i=1}^m \exp(\theta_i) \right) .$$

Note that $f^*(\mathbf{0}) = 0$. For each trial t define

$$\ell_t(\boldsymbol{\omega}) = \langle \boldsymbol{\omega}, M\mathbf{x}_t \rangle - \min_{\boldsymbol{\omega}' \in \Omega} \langle \boldsymbol{\omega}', M\mathbf{x}_t \rangle .$$

Thus $\inf_{\boldsymbol{\omega} \in \Omega} \ell_t(\boldsymbol{\omega}) = 0$. Note that $\partial_t = \{M\mathbf{x}_t\}$. For the specific choice of f from Eq. (31), and under the assumptions that all the entries in M are in $[-1, 1]$, it can be shown that for all $\boldsymbol{\theta}$ and t we have $\langle (M\mathbf{x}_t), \nabla^2 f^*(\boldsymbol{\theta})(M\mathbf{x}_t) \rangle \leq 1$ (see Appendix B). Thus, the conditions of Thm. 1 hold and we obtain the regret bound,

$$\forall \boldsymbol{\omega} \in \Omega, \quad \frac{1}{T} \sum_{t=1}^T \langle \boldsymbol{\omega}_t, M\mathbf{x}_t \rangle - \frac{1}{T} \sum_{t=1}^T \langle \boldsymbol{\omega}, M\mathbf{x}_t \rangle \leq \frac{f(\boldsymbol{\omega}) + \frac{1}{2}}{\sqrt{T}} . \quad (32)$$

Moreover, it can be verified that for all $\boldsymbol{\omega} \in \Omega$ we have $f(\boldsymbol{\omega}) \leq \log(m)$ and thus the right-hand side of the above becomes $(\log(m) + \frac{1}{2})/\sqrt{T}$.

If using the update given in Eq. (7), the resulted algorithm is similar to the algorithm described in [5]. Our regret bound however is slightly better. When defining $f(\boldsymbol{\omega}) = \frac{1}{2}(\|\boldsymbol{\omega}\|_2^2 - 1/m)$ the resulting algorithm is similar to an algorithm defined by [15]. However, the regret bound of this algorithm is significantly worse than the bound in Eq. (32).