Online Learning meets Optimization in the Dual

Shai Shalev-Shwartz[1] and Yoram Singer[1,2]

[1] School of Computer Sci. & Eng., The Hebrew University, Jerusalem 91904, Israel
[2] Google Inc., 1600 Amphitheater Parkway, Mountain View, CA 94043, USA
{shais,singer}@cs.huji.ac.il

Abstract. We describe a novel framework for the design and analysis of online learning algorithms based on the notion of duality in constrained optimization. We cast a sub-family of universal online bounds as an optimization problem. Using the weak duality theorem we reduce the process of online learning 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. We are thus able to tie the primal objective value and the number of prediction mistakes using and the increase in the dual. The end result is a general framework for designing and analyzing old and new online learning algorithms in the mistake bound model.

1 Introduction

Online learning of linear classifiers is an important and well-studied domain in machine learning with interesting theoretical properties and practical applications [3, 4, 7–10, 12]. An online learning algorithm observes instances in a sequence of trials. After each observation, the algorithm predicts a yes/no $(+/-)$ outcome. The prediction of the algorithm is formed by a hypothesis, which is a mapping from the instance space into $\{+1, -1\}$. This hypothesis is chosen by the online algorithm from a predefined class of hypotheses. Once the algorithm has made a prediction, it receives the correct outcome. Then, the online algorithm may choose another hypothesis from the class of hypotheses, presumably improving the chance of making an accurate prediction on subsequent trials. The quality of an online algorithm is measured by the number of prediction mistakes it makes along its run.

In this paper we introduce a general framework for the design and analysis of online learning algorithms. Our framework emerges from a new view on relative mistake bounds [10, 14], which are the common thread in the analysis of online learning algorithms. A relative mistake bound measures the performance of an online algorithm relatively to the performance of a competing hypothesis. The competing hypothesis can be chosen in hindsight from a class of hypotheses, after observing the entire sequence of examples. For example, the original mistake bound of the Perceptron algorithm [15], which was first suggested over 50 years ago, was derived by using a competitive analysis, comparing the algorithm to a linear hypothesis which achieves a large margin on the sequence of examples. Over the years, the competitive analysis technique was refined and extended to numerous prediction problems by employing complex and varied notions of progress toward a good competing hypothesis. The flurry of online learning algorithms sparked unified analyses of seemingly different online algorithms by Littlestone, Warmuth, Kivinen and colleagues [10, 13]. Most notably is the work of Grove, Littlestone, and Schuurmans [8] on a quasi-additive family of algorithms, which includes both the Perceptron [15] and the Winnow [13] algorithms as special cases. A similar unified view for regression was derived by Kivinen and Warmuth [10, 11]. Online algorithms for linear hypotheses and their analyses became more general and powerful by employing Bregman divergences for measuring the progress toward a good hypothesis [7–9]. In the aftermath of this paper we refer to these analyses as *primal* views.

We propose an alternative view of relative mistake bounds which is based on the notion of duality in constrained optimization. Online mistake bounds are universal in the sense that they hold for any possible predictor in a given hypothesis class. We therefore cast the universal bound as an optimization problem. Specifically, the objective function we cast is the sum of an empirical loss of a predictor and a complexity term for that predictor. The best predictor in a given class of hypotheses, which can only be determined in hindsight, is the minimizer of the optimization problem. In order to derive explicit quantitative mistake bounds we make an immediate use of the fact that dual objective lower bounds the primal objective. We therefore switch to the dual representation of the optimization problem. We then reduce the process of online learning 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 number of prediction mistakes, and the increase in the dual. The end result is a general framework for designing online algorithms and analyzing them in the mistake bound model.

We illustrate the power of our framework by studying two schemes for increasing the dual objective. The first performs a fixed size update based solely on the last observed example. We show that this dual update is equivalent to the primal update of the quasi-additive family of algorithms [8]. In particular, our framework yields the tightest known bounds for several known quasi-additive algorithms such as the Perceptron and Balanced Winnow. The second update scheme we study moves further in the direction of optimization techniques in several accounts. In this scheme the online learning algorithm may modify its hypotheses based on *multiple* past examples. Furthermore, the update itself is constructed by maximizing or approximately maximizing the increase in the dual. While this second approach still entertains the same mistake bound of the first scheme it also serves as a vehicle for deriving new online algorithms.

2 Problem Setting

In this section we introduce the notation used throughout the paper and formally describe our problem setting. 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 set of non-negative real numbers is denoted by $\mathbb{R}_+$. For any $k \geq 1$, the set of integers $\{1, \ldots, k\}$ is denoted by $[k]$.

Online learning of binary classifiers is performed in a sequence of trials. At trial t the algorithm first receives an instance $\mathbf{x}_t \in \mathbb{R}^n$ and is required to predict the label associated with that instance. We denote the prediction of the algorithm on the t'th trial by $\hat{y}_t$. For simplicity and concreteness we focus on online learning of binary classifiers, namely, we assume that the labels are in $\{+1, -1\}$. After the online learning algorithm has predicted the label $\hat{y}_t$, the true label $y_t \in \{+1, -1\}$ is revealed and the algorithm pays a unit cost if its prediction is wrong, that is, if $y_t \neq \hat{y}_t$. The ultimate goal of the algorithm is to minimize the total number of prediction mistakes it makes along its run. To achieve this goal, the algorithm may update its prediction mechanism after each trial so as to be more accurate in later trials.

In this paper, we assume that the prediction of the algorithm at each trial is determined by a margin-based linear hypothesis. Namely, there exists a weight vector $\boldsymbol{\omega}_t \in \Omega \subset \mathbb{R}^n$ where $\hat{y}_t = \text{sign}(\langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle)$ is the actual binary prediction and $|\langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle|$ is the confidence in this prediction. The term $y_t \langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle$ is called the *margin* of the prediction and is positive whenever y_t and $\text{sign}(\langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle)$ agree. We can evaluate the performance of a weight vector $\boldsymbol{\omega}$ on a given example $(\mathbf{x}, y)$ in one of two ways. First, we can check whether $\boldsymbol{\omega}$ results in a prediction mistake which amounts to checking whether $y = \text{sign}(\langle \boldsymbol{\omega}, \mathbf{x} \rangle)$ or not. Throughout this paper, we use M to denote the number of prediction mistakes made by an online algorithm on a sequence of examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$. The second way we evaluate the predictions of an hypothesis is by using the *hinge-loss* function, defined as,

$$\ell^{\gamma}\big(\boldsymbol{\omega}; (\mathbf{x}, y)\big) \;=\; \begin{cases} 0 & \text{if } \; y \langle \boldsymbol{\omega}, \mathbf{x} \rangle \geq \gamma \\ \gamma - y \langle \boldsymbol{\omega}, \mathbf{x} \rangle & \text{otherwise} \end{cases} \quad . \tag{1}$$

The hinge-loss penalizes an hypothesis for any margin less than γ. Additionally, if $y \neq \text{sign}(\langle \boldsymbol{\omega}, \mathbf{x} \rangle)$ then $\ell^{\gamma}(\boldsymbol{\omega}; (\mathbf{x}, y)) \geq \gamma$. Therefore, the *cumulative hinge-loss* suffered over a sequence of examples upper bounds γM. Throughout the paper, when $\gamma = 1$ we use the shorthand $\ell(\boldsymbol{\omega}; (\mathbf{x}, y))$.

As mentioned before, the performance of an online learning algorithm is measured by the cumulative number of prediction mistakes it makes along its run on a sequence of examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$. Ideally, we would like to think of the labels as if they are generated by an unknown yet *fixed* weight vector $\boldsymbol{\omega}^{\star}$ such that $y_i = \text{sign}(\langle \boldsymbol{\omega}^{\star}, \mathbf{x}_i \rangle)$ for all $i \in [m]$. Moreover, in an utopian case, the cumulative hinge-loss of $\boldsymbol{\omega}^{\star}$ on the entire sequence is zero, which means that $\boldsymbol{\omega}^{\star}$ produces the correct label with a confidence of at least γ. In this case, we would like M, the number of prediction mistakes of our online algorithm, to be independent of m, the number of examples. Usually, in such cases, M is upper bounded by $F(\boldsymbol{\omega}^{\star})$ where $F : \Omega \to \mathbb{R}$ is a function which measures the complexity of $\boldsymbol{\omega}^{\star}$. In the more realistic case, there does not exist $\boldsymbol{\omega}^{\star}$ which perfectly predicts the data. In this case, we would like the online algorithm to be competitive with *any* fixed hypothesis $\boldsymbol{\omega}$. Formally, let λ and C be two positive scalars. We say that our online algorithm is (λ, C)-competitive with the set of vectors in Ω, with respect to a complexity function F and the hinge-loss ℓ^{γ}, if the following bound holds,

$$\forall\, \boldsymbol{\omega} \in \Omega, \quad \lambda M \;\leq\; F(\boldsymbol{\omega}) + C \sum_{i=1}^{m} \ell^{\gamma}(\boldsymbol{\omega}; (\mathbf{x}_i, y_i)) \;\; . \tag{2}$$

The parameter C controls the trade-off between the complexity of $\boldsymbol{\omega}$ (through F) and the cumulative hinge-loss of $\boldsymbol{\omega}$. The parameter λ is introduced for technical reasons that are provided in the next section. The main goal of this paper is to develop a general paradigm for designing online learning algorithms and analyze them in the mistake bound framework given in Eq. (2).

3 A primal-dual apparatus for online learning

In this section we describe a methodology for designing online learning algorithms for binary classification. To motivate our construction let us first consider the special case where $\gamma = 1$, $F(\boldsymbol{\omega}) = \frac{1}{2}\|\boldsymbol{\omega}\|_2^2$, and $\Omega = \mathbb{R}^n$. Denote by $\mathcal{P}(\boldsymbol{\omega})$ the right hand side of Eq. (2) which in this special case amounts to,

$$\mathcal{P}(\boldsymbol{\omega}) \ = \ \frac{1}{2}\|\boldsymbol{\omega}\|^2 + C\sum_{i=1}^{m} \ell(\boldsymbol{\omega}; (\mathbf{x}_i, y_i)) \ .$$

The bound in Eq. (2) can be rewritten as,

$$\lambda\, M \ \leq \ \min_{\boldsymbol{\omega}\in\mathbb{R}^n} \ \mathcal{P}(\boldsymbol{\omega}) \ \overset{\text{def}}{=} \ \mathcal{P}^\star \ . \qquad (3)$$

Note that $\mathcal{P}(\boldsymbol{\omega})$ is the well-known primal objective function of the optimization problem employed by the SVM algorithm [5]. Intuitively, we view the online learning task as incrementally solving the optimization problem $\min_{\boldsymbol{\omega}} \mathcal{P}(\boldsymbol{\omega})$. However, while $\mathcal{P}(\boldsymbol{\omega})$ depends on the entire sequence of examples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$, the online algorithm is confined to use on trial t only the first $t-1$ examples of the sequence. To overcome this disparity, we follow the approach that ostriches take in solving problems: we simply ignore the examples $\{(\mathbf{x}_t, y_t), \ldots, (\mathbf{x}_m, y_m)\}$ as they are not provided to the algorithm on trial t. Therefore, on trial t we use the following weight vector for predicting the label,

$$\boldsymbol{\omega}_t \ = \ \underset{\boldsymbol{\omega}}{\operatorname{argmin}} \ \mathcal{P}_t(\boldsymbol{\omega}) \quad \text{where} \quad \mathcal{P}_t(\boldsymbol{\omega}) \ = \ \frac{1}{2}\|\boldsymbol{\omega}\|^2 + C\sum_{i=1}^{t-1} \ell(\boldsymbol{\omega}; (\mathbf{x}_i, y_i)) \ .$$

This online algorithm is a simple (and non-efficient) adaptation of the SVM algorithm for the online setting and we therefore call it the Online-SVM algorithm (see also [12]). Since the hinge-loss $\ell(\boldsymbol{\omega}; (\mathbf{x}_t, y_t))$ is non-negative we get that $\mathcal{P}_t(\boldsymbol{\omega}) \leq \mathcal{P}_{t+1}(\boldsymbol{\omega})$ for any $\boldsymbol{\omega}$ and therefore $\mathcal{P}_t(\boldsymbol{\omega}_t) \leq \mathcal{P}_t(\boldsymbol{\omega}_{t+1}) \leq \mathcal{P}_{t+1}(\boldsymbol{\omega}_{t+1})$. Note that $\mathcal{P}_1(\boldsymbol{\omega}_1) = 0$ and that $\mathcal{P}_{m+1}(\boldsymbol{\omega}) = \mathcal{P}^\star$. Thus,

$$0 \ = \ \mathcal{P}_1(\boldsymbol{\omega}_1) \ \leq \ \mathcal{P}_2(\boldsymbol{\omega}_2) \ \leq \ \ldots \ \leq \ \mathcal{P}_{m+1}(\boldsymbol{\omega}_{m+1}) \ = \ \mathcal{P}^\star \ .$$

Recall that our goal is to find an online algorithm which entertains the mistake bound given in Eq. (3). Suppose that we can show that for each trial t on which the online algorithm makes a prediction mistake we have that $\mathcal{P}_{t+1}(\boldsymbol{\omega}_{t+1}) - \mathcal{P}_t(\boldsymbol{\omega}_t) \ \geq \ \lambda \ > \ 0$. Equipped with this assumption, it follows immediately that if the online algorithm made M prediction mistakes on the entire sequence of examples then $\mathcal{P}_{m+1}(\boldsymbol{\omega}_{m+1})$ should be at least $\lambda\, M$. Since $\mathcal{P}_{m+1}(\boldsymbol{\omega}_{m+1}) = \mathcal{P}^\star$ we conclude that $\lambda\, M \leq \mathcal{P}^\star$ which gives the desired mistake bound from Eq. (3). In summary, to prove a mistake bound one needs to show that the online algorithm constructs a sequence of lower bounds $\mathcal{P}_1(\boldsymbol{\omega}_1), \ldots, \mathcal{P}_{m+1}(\boldsymbol{\omega}_{m+1})$ for $\mathcal{P}^\star$. These lower bounds should become tighter and tighter with the progress of the online algorithm. Moreover, whenever the algorithm makes a prediction mistake the lower bound must increase by at least λ.

The notion of duality, commonly used in optimization theory, plays an important role in obtaining lower bounds for the minimal value of the primal objective (see for example [2]). We now take an alternative view of the Online-SVM algorithm based on the notion of duality. As we formally show later, the dual of the problem $\min_{\boldsymbol{\omega}} \mathcal{P}(\boldsymbol{\omega})$ is

$$\max_{\boldsymbol{\alpha} \in [0,C]^m} \mathcal{D}(\boldsymbol{\alpha}) \quad \text{where } \mathcal{D}(\boldsymbol{\alpha}) \;=\; \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \left\| \sum_{i=1}^{m} \alpha_i \, y_i \, \mathbf{x}_i \right\|^2 \quad . \tag{4}$$

The weak duality theorem states that any value of the dual objective is upper bounded by the optimal primal objective. That is, for any $\boldsymbol{\alpha} \in [0, C]^m$ we have that $\mathcal{D}(\boldsymbol{\alpha}) \le \mathcal{P}^\star$. If in addition strong duality holds then $\max_{\boldsymbol{\alpha} \in [0,C]^m} \mathcal{D}(\boldsymbol{\alpha}) = \mathcal{P}^\star$. As we show in the sequel, the values $\mathcal{P}_1(\boldsymbol{\omega}_1), \ldots, \mathcal{P}_{m+1}(\boldsymbol{\omega}_{m+1})$ translate to a sequence of dual objective values. Put another way, there exists a sequence of dual solutions $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{m+1}$ such that for all $t \in [m+1]$ we have that $\mathcal{D}(\boldsymbol{\alpha}_t) = \mathcal{P}_t(\boldsymbol{\omega}_t)$. This fact follows from a property of the dual function in Eq. (4) as we now show.

Denote by $\mathcal{D}_t$ the dual objective function of $\mathcal{P}_t$,

$$\mathcal{D}_t(\boldsymbol{\alpha}) \;=\; \sum_{i=1}^{t-1} \alpha_i - \frac{1}{2} \left\| \sum_{i=1}^{t-1} \alpha_i \, y_i \, \mathbf{x}_i \right\|^2 \quad . \tag{5}$$

Note that $\mathcal{D}_t$ is a mapping from $[0, C]^{t-1}$ into the reals. From strong duality we know that the minimum of $\mathcal{P}_t$ equals to the maximum of $\mathcal{D}_t$. From the definition of $\mathcal{D}_t$ we get that for $(\alpha_1, \ldots, \alpha_{t-1}) \in [0, C]^{t-1}$ the following equality holds,

$$\mathcal{D}_t((\alpha_1, \ldots, \alpha_{t-1})) \;=\; \mathcal{D}((\alpha_1, \ldots, \alpha_{t-1}, 0, \ldots, 0)) \quad .$$

Therefore, the Online-SVM algorithm can be viewed as an incremental solver of the *dual* problem, $\max_{\boldsymbol{\alpha} \in [0,C]^m} \mathcal{D}(\boldsymbol{\alpha})$, where at the end of trial t the algorithm maximizes the dual function confined to the first t variables,

$$\max_{\boldsymbol{\alpha} \in [0,C]^m} \mathcal{D}(\boldsymbol{\alpha}) \qquad \text{s.t.} \;\; \forall i > t, \; \alpha_i = 0 \quad .$$

The property of the dual objective that we utilize is that it can be optimized in a sequential manner. Specifically, if on trial t we ground α_i to zero for $i \ge t$ then $\mathcal{D}(\boldsymbol{\alpha})$ does not depend on examples which have not been observed yet.

We presented two views of the Online-SVM algorithm. In the first view the algorithm constructs a sequence of *primal* solutions $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_{m+1}$ while in the second the algorithm constructs a sequence of *dual* solutions which we analogously denote by $\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^{m+1}$. As we show later, the connection between $\boldsymbol{\omega}_t$ and $\boldsymbol{\alpha}^t$ is given through the equality,

$$\boldsymbol{\omega}_t \;=\; \sum_{i=1}^{m} \alpha_i^t \, y_i \, \mathbf{x}_i \quad . \tag{6}$$

In general, any sequence of feasible dual solutions $\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^{m+1}$ can define an online learning algorithm by setting $\boldsymbol{\omega}_t$ according to Eq. (6). Naturally, we require that $\alpha_i^t = 0$ for all $i \geq t$ since otherwise $\boldsymbol{\omega}_t$ would depend on examples which have not been observed yet. To prove that the resulting online algorithm entertains the mistake bound given in Eq. (3) we impose two additional conditions. First, we require that $\mathcal{D}(\boldsymbol{\alpha}^{t+1}) \geq \mathcal{D}(\boldsymbol{\alpha}^t)$ which means that the dual objective never decreases. In addition, on trials in which the algorithm makes a prediction mistake we require that the increase of the dual objective will be strictly positive, $\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t) \geq \lambda > 0$. To recap, any incremental solver for the dual optimization problem which satisfies the above requirements can serve as an online algorithm which meets the mistake bound given in Eq. (3).

Let us now formally generalize the above motivating discussion. Our starting point is the desired mistake bound of the form given in Eq. (2), which can be rewritten as,

$$\lambda\, M \;\leq\; \inf_{\boldsymbol{\omega} \in \Omega} \left(F(\boldsymbol{\omega}) + C \sum_{i=1}^{m} \ell^{\gamma}(\boldsymbol{\omega}; (\mathbf{x}_i, y_i)) \right) \;. \tag{7}$$

As in our motivating example we denote by $\mathcal{P}(\boldsymbol{\omega})$ the primal objective of the optimization problem on the right-hand side of Eq. (7). Our goal is to develop an online learning algorithm that achieves this mistake bound. First, let us derive the dual optimization problem. Using the definition of ℓ^{γ} we can rewrite the optimization problem as,

$$\begin{aligned} &\inf_{\boldsymbol{\omega} \in \Omega, \boldsymbol{\xi} \in \mathbb{R}_+^m} \; F(\boldsymbol{\omega}) + C \sum_{i=1}^{m} \xi_i \\ &\quad \text{s.t. } \forall i \in [m], \;\; y_i \langle \boldsymbol{\omega}, \mathbf{x}_i \rangle \geq \gamma - \xi_i \;\; . \end{aligned} \tag{8}$$

We further rewrite this optimization problem using the Lagrange dual function,

$$\inf_{\boldsymbol{\omega} \in \Omega, \boldsymbol{\xi} \in \mathbb{R}_+^m} \; \sup_{\boldsymbol{\alpha} \in \mathbb{R}_+^m} \; \underbrace{F(\boldsymbol{\omega}) + C \sum_{i=1}^{m} \xi_i + \sum_{i=1}^{m} \alpha_i \left(\gamma - y_i \langle \boldsymbol{\omega}, \mathbf{x}_i \rangle - \xi_i \right)}_{\stackrel{\text{def}}{=} \mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\xi}, \boldsymbol{\alpha})} \;. \tag{9}$$

Eq. (9) is equivalent to Eq. (8) due to the following fact. If the constraint $y_i \langle \boldsymbol{\omega}, \mathbf{x}_i \rangle \geq \gamma - \xi_i$ holds then the optimal value of α_i in Eq. (9) is zero. If on the other hand the constraint does not hold then α_i equals ∞, which implies that $\boldsymbol{\omega}$ cannot constitute the optimal primal solution. The weak duality theorem (see for example [2]) states that,

$$\sup_{\boldsymbol{\alpha} \in \mathbb{R}_+^m} \; \inf_{\boldsymbol{\omega} \in \Omega, \boldsymbol{\xi} \in \mathbb{R}_+^m} \; \mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\xi}, \boldsymbol{\alpha}) \;\leq\; \inf_{\boldsymbol{\omega} \in \Omega, \boldsymbol{\xi} \in \mathbb{R}_+^m} \; \sup_{\boldsymbol{\alpha} \in \mathbb{R}_+^m} \; \mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\xi}, \boldsymbol{\alpha}) \;. \tag{10}$$

The dual objective function is defined to be,

$$\mathcal{D}(\boldsymbol{\alpha}) \;=\; \inf_{\boldsymbol{\omega} \in \Omega, \boldsymbol{\xi} \in \mathbb{R}_+^m} \; \mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\xi}, \boldsymbol{\alpha}) \;. \tag{11}$$

Using the definition of $\mathcal{L}$, we can rewrite the dual objective as a sum of three terms,

$$\mathcal{D}(\boldsymbol{\alpha}) \;=\; \gamma \sum_{i=1}^{m} \alpha_i \;-\; \sup_{\boldsymbol{\omega} \in \Omega} \left(\langle \boldsymbol{\omega}, \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \rangle - F(\boldsymbol{\omega}) \right) \;+\; \inf_{\boldsymbol{\xi} \in \mathbb{R}_+^m} \; \sum_{i=1}^{m} \xi_i \left(C - \alpha_i \right) \;.$$

The last term equals to zero for $\alpha_i \in [0, C]$ and to $-\infty$ for $\alpha_i > C$. Since our goal is to maximize $\mathcal{D}(\boldsymbol{\alpha})$ we can confine ourselves to the case $\boldsymbol{\alpha} \in [0, C]^m$ and simply write,

$$\mathcal{D}(\boldsymbol{\alpha}) \; = \; \gamma \sum_{i=1}^{m} \alpha_i \; - \; \sup_{\boldsymbol{\omega} \in \Omega} \left(\langle \boldsymbol{\omega}, \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \rangle - F(\boldsymbol{\omega}) \right) \quad .$$

The second term in the above presentation of $\mathcal{D}(\boldsymbol{\alpha})$ can be rewritten using the notion of conjugate functions (see for example [2]). Formally, the conjugate[3] of the function F is the function,

$$G(\boldsymbol{\theta}) \; = \; \sup_{\boldsymbol{\omega} \in \Omega} \; \langle \boldsymbol{\omega}, \boldsymbol{\theta} \rangle - F(\boldsymbol{\omega}) \quad . \qquad (12)$$

Using the definition of G we conclude that for $\boldsymbol{\alpha} \in [0, C]^m$ the dual objective function can be rewritten as,

$$\mathcal{D}(\boldsymbol{\alpha}) \; = \; \gamma \sum_{i=1}^{m} \alpha_i \; - \; G\left(\sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \right) \quad . \qquad (13)$$

For instance, it is easy to verify that the conjugate of $F(\boldsymbol{\omega}) = \frac{1}{2}\|\boldsymbol{\omega}\|_2^2$ (with $\Omega = \mathbb{R}^n$) is $G(\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{\theta}\|^2$. Indeed, the above definition of $\mathcal{D}$ for this case coincides with the value of $\mathcal{D}$ given in Eq. (4).

We now describe a template algorithm for online classification by incrementally increasing the dual objective function. Our algorithm starts with the trivial dual solution $\boldsymbol{\alpha}^1 = \mathbf{0}$. On trial t, we use $\boldsymbol{\alpha}^t$ for defining the weight vector $\boldsymbol{\omega}_t$ which is used for predicting the label as follows. First, we define $\boldsymbol{\theta}_t = \sum_{i=1}^{t-1} \alpha_i^t \, y_i \, \mathbf{x}_i$. Throughout the paper we assume that the supremum in the definition of $G(\boldsymbol{\theta})$ is attainable and set,

$$\boldsymbol{\omega}_t \; = \; \operatorname*{argmax}_{\boldsymbol{\omega} \in \Omega} \; \left(\langle \boldsymbol{\omega}, \boldsymbol{\theta}_t \rangle - F(\boldsymbol{\omega}) \right) \quad . \qquad (14)$$

Next, we use $\boldsymbol{\omega}_t$ for predicting the label $\hat{y}_t = \text{sign}(\langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle)$. Finally, we find a new dual solution $\boldsymbol{\alpha}^{t+1}$ with the last $m - t$ elements of $\boldsymbol{\alpha}^{t+1}$ are still grounded to zero. The two requirements we imposed imply that the new value of the dual objective, $\mathcal{D}(\boldsymbol{\alpha}^{t+1})$, should be at least $\mathcal{D}(\boldsymbol{\alpha}^t)$. Moreover, if we make a prediction mistake the increase in the dual objective should be strictly positive. In general, we might not be able to guarantee a minimal increase of the dual objective. In the next section we propose sufficient conditions which guarantee a minimal increase of the dual objective whenever the algorithm makes a prediction mistake. Our template algorithm is summarized in Fig. 1.

We conclude this section with a general mistake bound for online algorithms belonging to our framework. We need first to introduce some additional notation. Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ be a sequence of examples and assume that an online algorithm which is derived from the template algorithm is run on this sequence. We denote by $\mathcal{E}$ the set of trials on which the algorithm made a prediction mistake, $\mathcal{E} = \{t \in [m] : \hat{y}_t \neq y_t\}$. To remind the reader, the number of prediction mistakes of the algorithm is M and

[3] The function G is also called the Fenchel conjugate of F. In cases where F is differentiable with an invertible gradient, G is also called the Legendre transform of F.

```
INPUT: Regularization function $F(\boldsymbol{\omega})$ with domain $\Omega$ ;
       Trade-off Parameter $C$ ; hinge-loss parameter $\gamma$
INITIALIZE: $\boldsymbol{\alpha}^1 = \mathbf{0}$
For $t = 1, 2, \ldots, m$
    define $\boldsymbol{\omega}_t = \operatorname*{argmax}_{\boldsymbol{\omega} \in \Omega} \langle \boldsymbol{\omega}, \boldsymbol{\theta}_t \rangle - F(\boldsymbol{\omega})$ where $\boldsymbol{\theta}_t = \sum_{i=1}^{t-1} \alpha_i^t \, y_i \, \mathbf{x}_i$
    receive an instance $\mathbf{x}_t$ and predict its label: $\hat{y}_t = \text{sign}(\langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle)$
    receive correct label $y_t$
    If $\hat{y}_t \neq y_t$
        find $\boldsymbol{\alpha}^{t+1} \in [0, C]^t \times \{0\}^{m-t}$ such that $\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t) > 0$
    Else
        find $\boldsymbol{\alpha}^{t+1} \in [0, C]^t \times \{0\}^{m-t}$ such that $\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t) \geq 0$
```

Fig. 1. The template algorithm for online classification

thus $M = |\mathcal{E}|$. Last, we denote by λ the *average* increase of the dual objective over the trials in $\mathcal{E}$,

$$\lambda \;=\; \frac{1}{|\mathcal{E}|} \sum_{t \in \mathcal{E}} \left(\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t) \right) \quad . \tag{15}$$

Recall that $F(\boldsymbol{\omega})$ is our complexity measure for the vector $\boldsymbol{\omega}$. A natural assumption on F is that $\min_{\boldsymbol{\omega} \in \Omega} F(\boldsymbol{\omega}) = 0$. The intuitive meaning of this assumption is that the complexity of the “simplest” hypothesis in Ω is zero. The following theorem provides a mistake bound for any algorithm which belongs to our framework.

Theorem 1. *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ be a sequence of examples. Assume that an online algorithm of the form given in Fig. 1 is run on this sequence with a function $F : \Omega \to \mathbb{R}$ which satisfies $\min_{\boldsymbol{\omega} \in \Omega} F(\boldsymbol{\omega}) = 0$. Then,*

$$\lambda \, M \;\leq\; \inf_{\boldsymbol{\omega} \in \Omega} \left(F(\boldsymbol{\omega}) + C \sum_{t=1}^{m} \ell^{\gamma}(\boldsymbol{\omega}; (\mathbf{x}_t, y_t)) \right) \quad ,$$

where λ is as defined in Eq. (15).

Proof. We prove the claim by bounding $\mathcal{D}(\boldsymbol{\alpha}^{m+1})$ from above and below. First, let us rewrite $\mathcal{D}(\boldsymbol{\alpha}^{m+1})$ as $\mathcal{D}(\boldsymbol{\alpha}^1) + \sum_{t=1}^{m} \left(\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t) \right)$. Recall that $\boldsymbol{\alpha}^1$ is the zero vector and therefore $\boldsymbol{\theta}_1 = \mathbf{0}$ which gives,

$$\mathcal{D}(\boldsymbol{\alpha}^1) \;=\; 0 - \max_{\boldsymbol{\omega} \in \Omega} (\langle \boldsymbol{\omega}, \mathbf{0} \rangle - F(\boldsymbol{\omega})) \;=\; \min_{\boldsymbol{\omega} \in \Omega} F(\boldsymbol{\omega}) \quad .$$

Thus, the assumption $\min_{\boldsymbol{\omega} \in \Omega} F(\boldsymbol{\omega}) = 0$ implies that $\mathcal{D}(\boldsymbol{\alpha}^1) = 0$. Since on each round $\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t) \geq 0$ we conclude that,

$$\mathcal{D}(\boldsymbol{\alpha}^{m+1}) \;\geq\; \sum_{t \in \mathcal{E}} \left(\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t) \right) \;=\; |\mathcal{E}| \, \lambda \quad .$$

This provides a lower bound on $\mathcal{D}(\boldsymbol{\alpha}^{m+1})$. The upper bound $\mathcal{D}(\boldsymbol{\alpha}^{m+1}) \leq \mathcal{P}^\star$ follows directly from the weak duality theorem. Comparing the upper and lower bounds concludes our proof. □

The bound in Thm. 1 becomes meaningless when λ is excessively small. In the next section we analyze a few known online algorithms. We show that these algorithms tacitly impose sufficient conditions on F and on the sequence of input examples. These conditions guarantee a minimal increase of the dual objective which result in mistake bounds for each algorithm.

4 Analysis of known online algorithms

In the previous section we introduced a template algorithm for online learning. In this section we analyze the family of quasi-additive online algorithms described in [8, 10, 11] using the newly introduced dual view. This family includes several known algorithms such as the Perceptron algorithm [15], Balanced-Winnow [8], and the family of p-norm algorithms [7]. Recall that we cast online learning as the problem of incrementally increasing the dual objective function given by Eq. (13). We show in this section that all quasi-additive online learning algorithms can be viewed as employing the same procedure for incrementing Eq. (13). The sole difference between the algorithms is the complexity function F which leads to different forms of the function G. We exploit this fact by providing a unified analysis and mistake bounds to all the above algorithms. The bounds we obtain are as tight as the bounds that were derived for each algorithm individually yet our proofs are simpler.

To guarantee an increase in the dual as given by Eq. (13) on erroneous trials we devise the following procedure. First, if on trial t the algorithm did not make a prediction mistake we do not change $\boldsymbol{\alpha}$ and thus set $\boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}^t$. If on trial t there was a prediction mistake, we change only the t'th component of $\boldsymbol{\alpha}$ and set it to C. Formally, for $t \in \mathcal{E}$ the new vector $\boldsymbol{\alpha}^{t+1}$ is defined as,

$$\alpha_i^{t+1} \;=\; \begin{cases} \alpha_i^t & \text{if } i \neq t \\ C & \text{if } i = t \end{cases} \tag{16}$$

This form of update implies that the components of $\boldsymbol{\alpha}$ are either zero or C.

Before we continue with the derivation and analysis of online algorithms, let us first provide sufficient conditions for the update given by Eq. (16) which guarantee a minimal increase of the dual objective for all $t \in \mathcal{E}$. Let $t \in \mathcal{E}$ be a trial on which $\boldsymbol{\alpha}$ was updated. From the definition of $\mathcal{D}(\boldsymbol{\alpha})$ we get that the change in the dual objective due to the update is,

$$\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t) \;=\; \gamma\, C - G(\boldsymbol{\theta}_t + C\, y_t \mathbf{x}_t) + G(\boldsymbol{\theta}_t) \;\; . \tag{17}$$

Throughout this section we assume that G is twice differentiable. (This assumption indeed holds for the algorithms we analyze.) We denote by $\boldsymbol{g}(\boldsymbol{\theta})$ the gradient of G at $\boldsymbol{\theta}$ and by $H(\boldsymbol{\theta})$ the Hessian of G, that is, the matrix of second order derivatives of G with respect to $\boldsymbol{\theta}$. We would like to note in passing that the vector function $\boldsymbol{g}(\cdot)$ is often referred to as the *link* function (see for instance [1, 7, 10, 11]).

Using Taylor expansion of G around $\boldsymbol{\theta}_t$, we get that there exists $\boldsymbol{\theta}$ for which,

$$G(\boldsymbol{\theta}_t + C\, y_t \mathbf{x}_t) \;=\; G(\boldsymbol{\theta}_t) + C\, y_t \left\langle \mathbf{x}_t, \boldsymbol{g}(\boldsymbol{\theta}_t) \right\rangle + \frac{1}{2}\, C^2 \left\langle \mathbf{x}_t, H(\boldsymbol{\theta})\, \mathbf{x}_t \right\rangle \;\; . \tag{18}$$

Plugging the above equation into Eq. (17) gives that,

$$\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^{t}) \;=\; C\left(\gamma - y_t \langle \mathbf{x}_t, \boldsymbol{g}(\boldsymbol{\theta}_t) \rangle\right) - \frac{1}{2}\, C^2 \left\langle \mathbf{x}_t, H(\boldsymbol{\theta})\, \mathbf{x}_t \right\rangle \;\; . \tag{19}$$

We next show that $\boldsymbol{\omega}_t = \boldsymbol{g}(\boldsymbol{\theta}_t)$ and therefore the second term in the right-hand of Eq. (18) is negative. Put another way, moving $\boldsymbol{\theta}_t$ infinitesimally in the direction of $y_t \mathbf{x}_t$ decreases G. We then cap the amount by which the second order term can influence the dual value. To show that $\boldsymbol{\omega}_t = \boldsymbol{g}(\boldsymbol{\theta}_t)$ note that from the definition of G and $\boldsymbol{\omega}_t$, we get that for all $\boldsymbol{\theta}$ the following holds,

$$G(\boldsymbol{\theta}_t) + \langle \boldsymbol{\omega}_t, \boldsymbol{\theta} - \boldsymbol{\theta}_t \rangle \;=\; \langle \boldsymbol{\omega}_t, \boldsymbol{\theta}_t \rangle - F(\boldsymbol{\omega}_t) + \langle \boldsymbol{\omega}_t, \boldsymbol{\theta} - \boldsymbol{\theta}_t \rangle \;=\; \langle \boldsymbol{\omega}_t, \boldsymbol{\theta} \rangle - F(\boldsymbol{\omega}_t) \;\; . \tag{20}$$

In addition, $G(\boldsymbol{\theta}) \;=\; \max_{\boldsymbol{\omega} \in \Omega} \langle \boldsymbol{\omega}, \boldsymbol{\theta} \rangle - F(\boldsymbol{\omega}) \;\geq\; \langle \boldsymbol{\omega}_t, \boldsymbol{\theta} \rangle - F(\boldsymbol{\omega}_t)$. Combining Eq. (20) with the last inequality gives the following,

$$G(\boldsymbol{\theta}) \geq G(\boldsymbol{\theta}_t) + \langle \boldsymbol{\omega}_t, \boldsymbol{\theta} - \boldsymbol{\theta}_t \rangle \;\; . \tag{21}$$

Since Eq. (21) holds for all $\boldsymbol{\theta}$ it implies that $\boldsymbol{\omega}_t$ is a sub-gradient of G. In addition, since G is differentiable its only possible sub-gradient at $\boldsymbol{\theta}_t$ is its gradient, $\boldsymbol{g}(\boldsymbol{\theta}_t)$, and thus $\boldsymbol{\omega}_t = \boldsymbol{g}(\boldsymbol{\theta}_t)$. The simple form of the update and the link between $\boldsymbol{\omega}_t$ and θ_t through $\boldsymbol{g}$ can be summarized as the following simple yet general quasi-additive update:

If $\hat{y}_t = y_t$ **Set** $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t$ and $\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t$
If $\hat{y}_t \neq y_t$ **Set** $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + C y_t \mathbf{x}_t$ and $\boldsymbol{\omega}_{t+1} = \boldsymbol{g}(\boldsymbol{\theta}_{t+1})$

Getting back to Eq. (19) we get that,

$$\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^{t}) \;=\; C\left(\gamma - y_t \langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle\right) - \frac{1}{2}\, C^2 \left\langle \mathbf{x}_t, H(\boldsymbol{\theta})\, \mathbf{x}_t \right\rangle \;\; . \tag{22}$$

Recall that we assume that $t \in \mathcal{E}$ and thus $y_t \langle \mathbf{x}_t, \boldsymbol{\omega}_t \rangle \leq 0$. In addition, we later on show that $\langle \mathbf{x}, H(\boldsymbol{\theta}) \mathbf{x} \rangle \leq 1$ for all $\mathbf{x} \in \Omega$ with the particular choices of G and under certain assumptions on the norm of $\mathbf{x}$. We therefore can state the following corollary.

Corollary 1. *Let G be a twice differentiable function whose domain is $\mathbb{R}^n$. Denote by H the Hessian of G and assume that for all $\boldsymbol{\theta} \in \mathbb{R}^n$ and for all $\mathbf{x}_t$ ($t \in \mathcal{E}$) we have that $\langle \mathbf{x}_t, H(\boldsymbol{\theta}) \mathbf{x}_t \rangle \leq 1$. Then, under the conditions of Thm. 1 the update given by Eq. (16) ensures that, $\lambda \;\geq\; \gamma\, C - \frac{1}{2} C^2$.*

Example 1 (Perceptron). The Perceptron algorithm [15] is derived from Eq. (16) by setting $F(\boldsymbol{\omega}) = \frac{1}{2}\|\boldsymbol{\omega}\|^2$, $\Omega = \mathbb{R}^n$, and $\gamma = 1$. To see this, note that the conjugate function of F for this choice is, $G(\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{\theta}\|^2$. Therefore, the gradient of G at $\boldsymbol{\theta}_t$ is $g(\boldsymbol{\theta}_t) = \boldsymbol{\theta}_t$, which implies that $\boldsymbol{\omega}_t = \boldsymbol{\theta}_t$. We thus obtain a scaled version of the well known Perceptron update, $\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t + C\, y_t\, \mathbf{x}_t$. Assume that $\|\mathbf{x}_t\|_2 \leq 1$ for all $t \in [m]$.

Since the Hessian of G is the identity matrix we get that, $\langle \mathbf{x}_t, H(\boldsymbol{\theta})\, \mathbf{x}_t\rangle = \langle \mathbf{x}_t, \mathbf{x}_t\rangle \leq 1$. Therefore, we obtain the following mistake bound,

$$(C - \frac{1}{2}C^2)\, M \;\leq\; \min_{\boldsymbol{\omega}\in\mathbb{R}^n} \frac{1}{2}\|\boldsymbol{\omega}\|^2 + C \sum_{i=1}^{m} \ell(\boldsymbol{\omega}; (\mathbf{x}_i, y_i)) \;\; . \tag{23}$$

Note the sequence of predictions of the Perceptron algorithm does not depend on the actual value of C so long as $C > 0$. Therefore, we can choose C so as to minimize the right hand side of Eq. (23) and rewrite,

$$\forall \boldsymbol{\omega} \in \mathbb{R}^n,\;\; M \;\leq\; \min_{C\in(0,2)} \left(\frac{1}{C(1-\frac{1}{2}C)}\right) \left(\frac{1}{2}\,\|\boldsymbol{\omega}\|^2 + C \sum_{i=1}^{m} \ell(\boldsymbol{\omega}; (\mathbf{x}_i, y_i))\right) \;\; ,$$

where the domain $(0, 2)$ for C ensures that the bound will not become vacuous. Solving the right-hand side of the above equation for C yields the following mistake bound,

$$M \;\leq\; L \,+\, \frac{1}{2}\|\boldsymbol{\omega}\|^2 \left(1 + \sqrt{1 \,+\, 4\, L/\|\boldsymbol{\omega}\|^2}\right) \;\; ,$$

where $L = \sum_{i=1}^{m} \ell(\boldsymbol{\omega}; (\mathbf{x}_i, y_i))$. The proof is omitted due to the lack of space and will be presented in a long version of the paper. We would like to note that this bound is identical to the best known mistake bound for the Perceptron algorithm (see for example [7]). However, our proof technique is vastly different and enables us to derive mistake bounds for new algorithms, as we show later on in Sec. 5.

Example 2 (Balanced Winnow). We now analyze a version of the Winnow algorithm called Balanced-Winnow [8] which is also closely related to the Exponentiated-Gradient algorithm [10]. For brevity we refer to the algorithm we analyze simply as Winnow. To derive the Winnow algorithm we choose, $F(\boldsymbol{\omega}) = \sum_{i=1}^{n} \omega_i \log\left(\frac{\omega_i}{1/n}\right)$, and $\mathit{\Omega} = \left\{\boldsymbol{\omega} \in \mathbb{R}^n_+ : \sum_{i=1}^{n} \omega_i = 1\right\}$. The function F is the relative entropy between the probability vector $\boldsymbol{\omega}$ and the uniform vector $(\frac{1}{n}, \ldots, \frac{1}{n})$. The relative entropy is non-negative and measures the entropic divergence between two distributions. It attains a value of zero whenever the two vectors are equal. Therefore, the minimum value of $F(\boldsymbol{\omega})$ is zero and is attained for $\boldsymbol{\omega} = (\frac{1}{n}, \ldots, \frac{1}{n})$. The conjugate of F is the logarithm of the sum of exponentials (see for example [2][pp. 93]), $G(\boldsymbol{\theta}) = \log\left(\sum_{i=1}^{n} e^{\theta_i}\right)$. The k'th element of the gradient of G is, $g_k(\boldsymbol{\theta}) = e^{\theta_k} / \left(\sum_{i=1}^{n} e^{\theta_i}\right)$. Note that $g(\boldsymbol{\theta})$ is a vector in the n-dimensional simplex and therefore $\boldsymbol{\omega}_t = g(\boldsymbol{\theta}_t) \in \mathit{\Omega}$. The k'th element of $\boldsymbol{\omega}_{t+1}$ can be rewritten using a multiplicative update rule,

$$\boldsymbol{\omega}_{t+1,k} \;=\; \frac{1}{Z_t} e^{\theta_{t,k} + C\, y_t\, \mathbf{x}_{t,k}} \;=\; \frac{1}{Z_t} e^{C\, y_t\, \mathbf{x}_{t,k}}\, \boldsymbol{\omega}_{t,k} \;\; ,$$

where Z_t is a normalization constant which ensures that $\boldsymbol{\omega}_{t+1}$ is in the simplex.

To analyze the algorithm we need to show that $\langle \mathbf{x}_t, H(\boldsymbol{\theta})\, \mathbf{x}_t\rangle \leq 1$, which indeed holds for $\|\mathbf{x}_t\|_\infty \leq 1$. The proof is omitted due to the lack of space. As a result, we obtain the following mistake bound,

$$\left(\gamma\, C - \frac{1}{2}C^2\right)\, M \;\leq\; \min_{\boldsymbol{\omega}\in\mathit{\Omega}} \left(\sum_{i=1}^{n} \omega_i \log(\omega_i) + \log(n) + C \sum_{i=1}^{m} \ell^{\gamma}(\boldsymbol{\omega}; (\mathbf{x}_i, y_i))\right) \;\; .$$

Since $\sum_{i=1}^{n} \omega_i \log(\omega_i) \leq 0$, if we set $C = \gamma$, the above bound reduces to,

$$M \;\leq\; 2 \left(\frac{\log(n)}{\gamma^2} \;+\; \min_{\boldsymbol{\omega} \in \Omega} \; \frac{1}{\gamma} \; \sum_{i=1}^{m} \ell^{\gamma}(\boldsymbol{\omega}; (\mathbf{x}_i, y_i)) \right) \;\; .$$

Example 3 (p-norm algorithms). We conclude this section with the analysis of the family of p-norm algorithms [7, 8]. Let $p, q \geq 1$ be two scalars such that $\frac{1}{p} + \frac{1}{q} = 1$. Define, $F(\boldsymbol{\omega}) \;=\; \frac{1}{2}\|\boldsymbol{\omega}\|_q^2 \;=\; \frac{1}{2}\left(\sum_{i=1}^{n} |\omega_i|^q\right)^{2/q}$, and let $\Omega = \mathbb{R}^n$. The conjugate function of F in this case is, $G(\boldsymbol{\theta}) \;=\; \frac{1}{2}\|\boldsymbol{\theta}\|_p^2$ (For a proof see [2], page 93.) and the i'th element of the gradient of G is,

$$g_i(\boldsymbol{\theta}) = \frac{\text{sign}(\theta_i)\,|\theta_i|^{p-1}}{\|\boldsymbol{\theta}\|_p^{p-2}} \;\; .$$

To analyze any p-norm algorithm we need again to bound for all t the quadratic form $\langle \mathbf{x}_t, H(\boldsymbol{\theta})\mathbf{x}_t \rangle$. It is possible to show (see [7, 8]) that

$$\langle \mathbf{x}\,,\; H(\boldsymbol{\theta})\,\mathbf{x} \rangle \;\leq\; \frac{1}{p}\left(\|\boldsymbol{\theta}\|_p^p\right)^{\frac{2}{p}-1}\; p\,(p-1) \sum_{i=1}^{n} \text{sign}(\theta_i)|\theta_i|^{p-2} x_i^2 \;\; . \qquad (24)$$

Using Holder inequality with the dual norms $\frac{p}{p-2}$ and $\frac{p}{2}$ we get that,

$$\sum_{i=1}^{n} \text{sign}(\theta_i)|\theta_i|^{p-2} x_i^2 \;\leq\; \left(\sum_{i=1}^{n} |\theta_i|^{(p-2)\frac{p}{p-2}}\right)^{\frac{p-2}{p}} \left(\sum_{i=1}^{n} {x_i^2}^{\frac{p}{2}}\right)^{\frac{2}{p}} \;=\; \|\boldsymbol{\theta}\|_p^{p-2}\,\|\mathbf{x}\|_p^2 \;\; .$$

Combining the above with Eq. (24) gives, $\langle \mathbf{x}\,,\; H(\boldsymbol{\theta})\,\mathbf{x} \rangle \;\leq\; (p-1)\|\mathbf{x}\|_p^2$. If we further assume that $\|\mathbf{x}\|_p \leq \sqrt{1/(p-1)}$ then we can apply corollary 1 and obtain that,

$$\left(\gamma\, C - \frac{1}{2}C^2\right)\, M \;\leq\; \min_{\boldsymbol{\omega} \in \mathbb{R}^n} \left(\frac{1}{2}\|\boldsymbol{\omega}\|_q^2 + C \sum_{i=1}^{m} \ell^{\gamma}(\boldsymbol{\omega}; (\mathbf{x}_i, y_i)) \right) \;\; .$$

5 Deriving new online learning algorithms

In the previous section we described a family of online learning algorithms. The algorithms are based on the simple procedure defined via Eq. (16) which increments the dual using a fixed-size update to a single dual variable. Intuitively, an update scheme which results in a larger increase in the dual objective on each trial is likely to yield online algorithms with refined loss bounds. In this section we outline a few new online update schemes which set $\boldsymbol{\alpha}$ more aggressively.

The update scheme of the previous section for increasing the dual modifies $\boldsymbol{\alpha}$ only on trials on which there was a prediction mistake ($t \in \mathcal{E}$). The update is performed by setting the t'th element of $\boldsymbol{\alpha}$ to C and keeping the rest of the variables intact. This simple update can be enhanced in several ways. First, note that while setting α_t^{t+1} to C guarantees a sufficient increase in the dual, there might be other values α_t^{t+1} which would lead to larger increases of the dual. Furthermore, we can also update $\boldsymbol{\alpha}$ on trials on which the prediction was correct so long as the loss is non-zero. Last, we need not restrict our update to the t'th element of $\boldsymbol{\alpha}$. We can instead update several dual variables as long as their indices are in $[t]$.

We now describe and briefly analyze a few new updates which increase the dual more aggressively. The goal here is to illustrate the power of the approach and the list of new updates we outline is by no means exhaustive. We start by describing an update which sets α_t^{t+1} adaptively, depending on the loss suffered on round t. This improved update constructs $\boldsymbol{\alpha}^{t+1}$ as follows,

$$\alpha_i^{t+1} = \begin{cases} \alpha_i^t & \text{if } \; i \neq t \\ \min\left\{\ell(\boldsymbol{\omega}_t; (\mathbf{x}_t, y_t))\,,\, C\right\} & \text{if } \; i = t \end{cases} \quad . \tag{25}$$

As before, the above update can be used with various complexity functions for F, yielding different quasi-additive algorithms. We now provide a unified analysis for all algorithms which are based on the update given by Eq. (25). In contrast to the previous update which modified $\boldsymbol{\alpha}$ only when there was a prediction mistake, the new update modifies $\boldsymbol{\alpha}$ whenever $\ell(\boldsymbol{\omega}_t; (\mathbf{x}_t, y_t)) > 0$. This more aggressive approach leads to a more general *loss* bound while still attaining the same mistake bound of the previous section. The mistake bound still holds since whenever the algorithm makes a prediction mistake its loss is at least γ. Formally, let us define the following mitigating function,

$$\mu(x) \;=\; \frac{1}{C}\left(\min\{x, C\}\left(x - \frac{1}{2}\min\{x, C\}\right)\right) \quad .$$

The function μ is illustrated in Fig. 2. Note that $\mu(\cdot)$ becomes very similar to the identity function for small values of C. The following theorem provides a bound on the cumulative sum of $\mu(\ell(\boldsymbol{\omega}_t, (\mathbf{x}_t, y_t)))$.

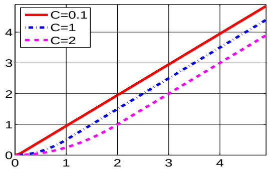

Fig. 2. The mitigating function $\mu(x)$ for different values of C.

Theorem 2. *Let* $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ *be a sequence of examples and let* $F : \Omega \to \mathbb{R}$ *be a complexity function for which* $\min_{\boldsymbol{\omega} \in \Omega} F(\boldsymbol{\omega}) = 0$. *Assume that an online algorithm is derived from Eq. (25) using* G *as the conjugate function of* F. *If* G *is twice differentiable and its Hessian satisfies,* $\langle \mathbf{x}_t, H(\boldsymbol{\theta})\mathbf{x}_t \rangle \leq 1$ *for all* $\boldsymbol{\theta} \in \mathbb{R}^n$ *and* $t \in [m]$, *then the following bound holds,*

$$\sum_{t=1}^{m} \mu\left(\ell(\boldsymbol{\omega}_t; (\mathbf{x}_t, y_t))\right) \;\leq\; \min_{\boldsymbol{\omega} \in \Omega}\left(\frac{1}{C}\, F(\boldsymbol{\omega}) + \sum_{t=1}^{m} \ell(\boldsymbol{\omega}; (\mathbf{x}_t, y_t))\right) \quad .$$

Proof. Analogously to the proof of Thm. 1, we prove this theorem by bounding $\mathcal{D}(\boldsymbol{\alpha}^{m+1})$ from above and below. The upper bound $\mathcal{D}(\boldsymbol{\alpha}^{m+1}) \leq \mathcal{P}^\star$ follows again from weak duality theorem. To derive a lower bound, note that the conditions stated in the theorem imply that $\mathcal{D}(\boldsymbol{\alpha}^1) = 0$ and thus $\mathcal{D}(\boldsymbol{\alpha}^{m+1}) = \sum_{t=1}^{m}\left(\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t)\right)$. Define $\tau_t = \min\{\ell(\boldsymbol{\omega}_t; (\mathbf{x}_t, y_t)), C\}$ and note that the sole difference between the updates given by Eq. (25) and Eq. (16) is that τ_t replaces C. Thus, the derivation of Eq. (22) in Sec. 4 can be repeated almost verbatim with τ_t replacing C to get,

$$\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t) \;\geq\; \tau_t\,\left(\gamma - y_t\langle \boldsymbol{\omega}_t, \mathbf{x}_t\rangle\right) - \frac{1}{2}\,\tau_t^2 \quad . \tag{26}$$

Summing over $t \in [m]$ and using the definitions of $\ell(\boldsymbol{\omega}_t; (\mathbf{x}_t, y_t))$, τ_t, and μ gives that,

$$\mathcal{D}(\boldsymbol{\alpha}^{m+1}) \;=\; \sum_{t=1}^{m} \left(\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^{t})\right) \;\geq\; C \sum_{t=1}^{m} \mu\left(\ell(\boldsymbol{\omega}_t; (\mathbf{x}_t, y_t))\right) \;\; .$$

Finally, we compare the lower and upper bounds on $\mathcal{D}(\boldsymbol{\alpha}^{m+1})$ and rearrange terms. □

Note that $\ell(\boldsymbol{\omega}_t; (\mathbf{x}_t, y_t)) \geq \gamma$ whenever the algorithm makes a prediction mistake. Since μ is a monotonically increasing function we get that the increase in the dual for $t \in \mathcal{E}$ is at least $\mu(\gamma)$. Thus, we obtain the mistake bound,

$$\lambda\, M \;\leq\; \mathcal{P}^\star \quad \text{where} \;\; \lambda \;\geq\; C\,\mu(\gamma) \;=\; \begin{cases} \gamma\, C - \frac{1}{2}\, C^2 & \text{if } \; C \leq \gamma \\ \frac{1}{2}\, \gamma^2 & \text{if } \; C > \gamma \end{cases} \quad . \tag{27}$$

The new update is advantageous over the previous update since in addition to the same increase in the dual on trials with a prediction mistake it is also guaranteed to increase the dual by $\mu(\ell(\cdot))$ on the rest of the trials. Yet, both updates are confined to modifying a single dual variable on each trial. We nonetheless can increase the dual more dramatically by modifying multiple dual variables on each round. Formally, for $t \in [m]$, let I_t be a subset of $[t]$ which includes t. Given I_t, we can set $\boldsymbol{\alpha}^{t+1}$ to be,

$$\boldsymbol{\alpha}^{t+1} \;=\; \operatorname*{argmax}_{\boldsymbol{\alpha} \in [0,C]^m} \mathcal{D}(\boldsymbol{\alpha}) \;\; \text{s.t.} \;\; \forall i \notin I_t, \; \alpha_i = \alpha_i^t \;\; . \tag{28}$$

This more general update also achieves the bound of Thm. 2 and the minimal increase in the dual as given by Eq. (27). To see this, note that the requirement that $t \in I_t$ implies,

$$\mathcal{D}(\boldsymbol{\alpha}^{t+1}) \;\geq\; \max\left\{\mathcal{D}(\boldsymbol{\alpha}) \;:\; \boldsymbol{\alpha} \in [0,C]^m \;\text{ and }\; \forall i \neq t, \; \alpha_i = \alpha_i^t\right\} \;\; . \tag{29}$$

Thus the increase in the dual $\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t)$ is guaranteed to be at least as large as the increase due to the previous updates. The rest of the proof of the bound is literally the same.

Let us now examine a few choices for I_t. Setting $I_t = [t]$ for all t gives the Online-SVM algorithm we mentioned in Sec. 3 by choosing $F(\boldsymbol{\omega}) = \frac{1}{2}\|\boldsymbol{\omega}\|^2$ and $\Omega = \mathbb{R}^n$. This algorithm makes use of all the examples that have been observed and thus is likely to make the largest increase in the dual objective on each trial. It does require however a full-blown quadratic programming solver. In contrast, Eq. (29) can be solved analytically when we employ the smallest possible set, $I_t = \{t\}$, with $F(\boldsymbol{\omega}) = \frac{1}{2}\|\boldsymbol{\omega}\|^2$. In this case α_t^{t+1} turns out to be the minimum between C and $\ell(\boldsymbol{\omega}_t; (\mathbf{x}_t, y_t))/\|\mathbf{x}_t\|^2$. This algorithm was described in [4] and belongs to a family of Passive Aggressive algorithms. The mistake bound that we obtain as a by product in this paper is however superior to the one in [4]. Naturally, we can interpolate between the minimal and maximal choices for I_t by setting the size of I_t to a predefined value k and choosing, say, the last k observed examples as the elements of I_t. For $k = 1$ and $k = 2$ we can solve Eq. (28) analytically while gaining modest increases in the dual. The full power of the update is unleashed for large values of k, however, Eq. (28) cannot be solved analytically and requires the usage of iterative procedures such as interior point methods.

6 Discussion

We presented a new framework for the design and analysis of online learning algorithms. Our framework yields the best known bounds for quasi-additive online classification algorithms. It also paves the way to new algorithms. There are various possible extensions of the work that we did not discuss due to the lack of space. Our framework can naturally be extended to other prediction problems such as regression, multiclass categorization, and ranking problems. Our framework is also applicable to settings where the target hypothesis is not fixed but rather drifting with the sequence of examples. In addition, the hinge-loss was used in our derivation in order to make a clear connection to the quasi-additive algorithms. The choice of the hinge-loss is rather arbitrary and it can be replaced with others such as the logistic loss. There are also numerous possible algorithmic extensions and new update schemes which manipulate multiple dual variables on each online update. Finally, our framework can be used with non-differentiable conjugate functions which might become useful in settings where there are combinatorial constraints on the number of non-zero dual variables (see [6]).

Acknowledgment This work was supported by grant 522-04 from the Israeli Science Foundation (ISF) and by grant I-773-8.6/2003 from the German Israeli Foundation (GIF).

References

1. K. Azoury and M. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.
2. S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
3. N. Cesa-Bianchi, A. Conconi, and C.Gentile. On the generalization ability of on-line learning algorithms. In *Advances in Neural Information Processing Systems 14*, pages 359–366, 2002.
4. K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive aggressive algorithms. Technical report, The Hebrew University, 2005.
5. N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.
6. O. Dekel, S. Shalev-Shwartz, and Y. Singer. The Forgetron: A kernel-based perceptron on a fixed budget. In *Advances in Neural Information Processing Systems 18*, 2005.
7. C. Gentile. The robustness of the p-norm algorithms. *Machine Learning*, 53(3), 2002.
8. A. J. Grove, N. Littlestone, and D. Schuurmans. General convergence results for linear discriminant updates. *Machine Learning*, 43(3):173–210, 2001.
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. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–64, January 1997.
11. J. Kivinen and M. Warmuth. Relative loss bounds for multidimensional regression problems. *Journal of Machine Learning*, 45(3):301–329, July 2001.
12. Y. Li and P. M. Long. The relaxed online maximum margin algorithm. *Machine Learning*, 46(1–3):361–387, 2002.
13. N. Littlestone. Learning when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

14. N. Littlestone. *Mistake bounds and logarithmic linear-threshold learning algorithms*. PhD thesis, U. C. Santa Cruz, March 1989.
15. F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958. (Reprinted in *Neurocomputing* (MIT Press, 1988).).