Mach Learn (2007) 69: 115–142
DOI 10.1007/s10994-007-5014-x

# A primal-dual perspective of online learning algorithms

**Shai Shalev-Shwartz · Yoram Singer**

Received: 21 September 2006 / Revised: 6 March 2007 / Accepted: 17 May 2007 /
Published online: 11 July 2007


**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 for analyzing online learning algorithms. We are thus able to tie the primal objective value and the number of prediction mistakes using the increase in the dual.

**Keywords** Online learning · Mistake bounds · Duality · Regret bounds

## 1 Introduction

Online learning of linear classifiers is an important and well-studied domain in machine learning with interesting theoretical properties and practical applications (Cesa-Bianchi et al. 2002; Crammer et al. 2005; Gentile 2001, 2002; Grove et al. 2001; Helmbold et al. 1999; Kivinen et al. 2002; Kivinen and Warmuth 1997; Li and Long 2002). 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

Editors: Hans Ulrich Simon, Gabor Lugosi, Avrim Blum.

A preliminary version of this paper appeared at the 19th Annual Conference on Learning Theory under the title "Online learning meets optimization in the dual".

S. Shalev-Shwartz (✉) · Y. Singer
School of Computer Science & Engineering, The Hebrew University, Jerusalem 91904, Israel
e-mail: shais@cs.huji.ac.il

Y. Singer
e-mail: singer@cs.huji.ac.il

Y. Singer
Google Inc., 1600 Amphitheater Parkway, Mountain View, CA 94043, USA

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 (Kivinen and Warmuth 1997; Littlestone 1989), 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 (Rosenblatt 1958), 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 techniques were 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 (Kivinen and Warmuth 1997; Littlestone 1988). Most notably is the work of Grove, Littlestone, and Schuurmans (Grove et al. 2001) on a quasi-additive family of algorithms, which includes both the Perceptron (Rosenblatt 1958) and the Winnow (Littlestone 1988) algorithms as special cases. A similar unified view for regression was derived by Kivinen and Warmuth (1997, 2001). 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 (Gentile 2002; Grove et al. 2001; Kivinen et al. 2002).

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 together the primal objective value and the number of prediction mistakes using the increase in the dual objective. 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 which is 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 (Grove et al. 2001). 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. Moreover, the update itself

is constructed by maximizing, or approximately maximizing, the increase in the dual. This second approach still entertains the same mistake bound of the first scheme. Moreover, it also serves as a vehicle for deriving new online algorithms which attain regret bounds with respect to the hinge-loss.

This paper is organized as follows. In Sect. 2 we begin with a formal presentation of online learning. Our new framework for designing and analyzing online learning algorithms is introduced in Sect. 3. Next, in Sect. 4, we derive the family of quasi-additive algorithms (Grove et al. 2001) by utilizing the newly introduced framework and show that our analysis produces the best known mistake bounds for these algorithms. In Sect. 5 we derive new online learning algorithms based on our framework. We analyze the performance of these algorithms in the mistake bound model as well as in the regret bound model in which the cumulative *loss* of the online algorithm is compared to the cumulative *loss* of any competing hypothesis. We recap and draw connections to earlier analysis techniques in Sect. 6. Possible extensions of our work and concluding remarks are given in Sect. 7.

## 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 $\omega$), 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 then 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 $\hat{y}_t = \mathrm{sign}(\langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle)$ 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 $\mathrm{sign}(\langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle)$ agree. We evaluate the performance of a weight vector $\boldsymbol{\omega}$ on a given example $(\mathbf{x}, y)$ in one of two ways. First, we may check whether the prediction based on $\boldsymbol{\omega}$ results in a mistake which amounts to checking whether $y = \mathrm{sign}(\langle \boldsymbol{\omega}, \mathbf{x} \rangle)$ 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,

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

The hinge-loss penalizes an hypothesis for any margin less than $\gamma$. Additionally, if $y \neq \mathrm{sign}(\langle \boldsymbol{\omega}, \mathbf{x} \rangle)$ then $\ell^{\gamma}(\boldsymbol{\omega}; (\mathbf{x}, y)) \geq \gamma$. Therefore, the *cumulative hinge-loss* suffered over a sequence of examples upper bounds $\gamma M$. 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 the utopian case where the cumulative hinge-loss of $\boldsymbol{\omega}^\star$ on the entire sequence is zero, the predictions that $\boldsymbol{\omega}^\star$ makes are all correct and with a confidence level of at least $\gamma$. 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 correctly predicts the labels of all observed instances. In this case, we would like the online algorithm to be competitive with *any* fixed hypothesis $\boldsymbol{\omega}$. Formally, let $\lambda$ and $C$ be two positive scalars. We say that our online algorithm is $(\lambda, C)$-competitive with the set of vectors in $\Omega$, with respect to a complexity function $F$ and the hinge-loss $\ell^\gamma$, 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 $\boldsymbol{\omega}$ (measured through $F$) and the cumulative hinge-loss of $\boldsymbol{\omega}$. The parameter $\lambda$ 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 (2).

## 3 A primal-dual view of online learning

In this section we describe our methodology for designing and analyzing online learning algorithms for binary classification problems. Let us first rewrite the bound in (2) as follows,

$$\lambda M \leq \min_{\boldsymbol{\omega} \in \Omega} \mathcal{P}(\boldsymbol{\omega}), \tag{3}$$

where $\mathcal{P}(\boldsymbol{\omega})$ denotes the right-hand side of (2). Let us also denote by $\mathcal{P}^\star$ the right-hand side of (3). To motivate our construction we start by analyzing a specific online learning algorithm, denoted *Follow-the-Regularized-Leader* or FoReL in short. 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 do this, the FoReL algorithm simply ignores the examples $\{(\mathbf{x}_t, y_t), \ldots, (\mathbf{x}_m, y_m)\}$ as they are not provided to the algorithm on trial $t$. Formally, let $\mathcal{P}_t(\boldsymbol{\omega})$ denote the following *instantaneous* objective function,

$$\mathcal{P}_t(\boldsymbol{\omega}) = F(\boldsymbol{\omega}) + C \sum_{i=1}^{t-1} \ell^\gamma(\boldsymbol{\omega}; (\mathbf{x}_i, y_i)).$$

The FoReL algorithm sets $\boldsymbol{\omega}_t$ to be the optimal solution of $\mathcal{P}_t(\boldsymbol{\omega})$ over $\boldsymbol{\omega} \in \Omega$. Since $\mathcal{P}_t(\boldsymbol{\omega})$ depends only on the sequence of examples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{t-1}, y_{t-1})\}$ it indeed adheres with the main requirement of an online algorithm. The role of this algorithm is to emphasize the

difficulties encountered in employing a *primal* algorithm and to pave the way to our approach which is based on the dual representation of the optimization problem $\min_{\boldsymbol{\omega}} \mathcal{P}(\boldsymbol{\omega})$. The FoReL algorithm can be viewed as a modification of the follow-the-leader algorithm, originally suggested by Hannan (1957). In contrast to follow-the-leader algorithms, our regularized version of the algorithm also takes the complexity of $\boldsymbol{\omega}$ in the form of $F(\boldsymbol{\omega})$ into account when constructing its predictors. We would like to note that in general follow-the-leader algorithms may not attain a mistake bound while under the assumptions outlined below the regularized version of follow-the-leader does yield a mistake bound. Before proceeding to the mistake bound analysis, we also would like to mention that when $F(\boldsymbol{\omega}) = \frac{1}{2}\|\boldsymbol{\omega}\|_2^2$ the algorithm reduces to a simple (and rather inefficient) adaptation of the SVM algorithm to an online setting (see also Li and Long 2002; Cesa-Bianchi et al. 2005; Vovk 2001). When the loss function is the squared-loss and the task is linear regression, the FoReL algorithm is similar to the well known online ridge regression algorithm.

We now turn to the analysis of the FoReL algorithm. First, we need to introduce additional notation. Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ be a sequence of examples and denote by $\mathcal{E}$ the set of trials on which the algorithm made a prediction mistake,

$$\mathcal{E} = \{t \in [m] : \operatorname{sign}(\langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle) \neq y_t\}. \tag{4}$$

To remind the reader, the number of prediction mistakes of the algorithm is denoted by $M$ and thus $M = |\mathcal{E}|$. To prove a bound of the form given in (3) we associate a scalar, denoted $v_t$, with each weight vector $\boldsymbol{\omega}_t$. Intuitively, the scalar $v_t$ measures the quality of $\boldsymbol{\omega}_t$ in predicting the labels. To ensure proper normalization of the quality assessment we require that the quality value of the initial weight vector is 0 and that the quality values of all weight vectors is at most $\mathcal{P}^\star$. The following lemma states that a sufficient condition for proving a mistake bound is that the sequence of quality values $v_1, \ldots, v_{m+1}$ corresponding to the weight vectors $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_{m+1}$ never decreases.

**Lemma 1** *Assume that an arbitrary online learning algorithm is presented with the sequence of examples* $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ *and let* $\mathcal{E}$ *be as defined in* (4). *Assume in addition that we can associate a scalar* $v_t$ *with each weight vector* $\boldsymbol{\omega}_t$ *constructed by the online algorithm such that the following requirements hold*:

$$\text{(i)} \quad v_1 = 0; \qquad \text{(ii)} \quad v_1 \leq v_2 \leq \cdots \leq v_{m+1}; \qquad \text{(iii)} \quad v_{m+1} \leq \mathcal{P}^\star.$$

*Then*, $\lambda M \leq \mathcal{P}^\star$ *where*

$$\lambda = \frac{1}{M} \sum_{t \in \mathcal{E}} (v_{t+1} - v_t).$$

*Proof* Combining the three requirements and using the definition of $\lambda$ give that

$$\mathcal{P}^\star \geq v_{m+1} = v_{m+1} - v_0 = \sum_{t=1}^{m} (v_{t+1} - v_t) \geq \sum_{t \in \mathcal{E}} (v_{t+1} - v_t) = M\lambda. \qquad \square$$

The above lemma underlines a method for obtaining mistake bounds by finding a sequence of quality values $v_1, \ldots, v_{m+1}$ each of which is associated with a weight vector used for prediction. These values should satisfy the conditions stated in the lemma in order to prove mistake bounds. We now follow this line of proof for analyzing the FoReL algorithm by defining $v_t = \mathcal{P}_t(\boldsymbol{\omega}_t)$.

Since the hinge-loss $\ell^{\gamma}(\boldsymbol{\omega}; (\mathbf{x}_t, y_t))$ is non-negative we get that for any vector $\boldsymbol{\omega}$, $\mathcal{P}_t(\boldsymbol{\omega}) \leq \mathcal{P}_{t+1}(\boldsymbol{\omega})$ and in particular $\mathcal{P}_t(\boldsymbol{\omega}_{t+1}) \leq \mathcal{P}_{t+1}(\boldsymbol{\omega}_{t+1})$. The optimality of each vector $\boldsymbol{\omega}_t$ with respect to $\mathcal{P}_t(\boldsymbol{\omega})$ implies that $\mathcal{P}_t(\boldsymbol{\omega}_t) \leq \mathcal{P}_t(\boldsymbol{\omega}_{t+1})$. Combining the last two inequalities we get that $\mathcal{P}_t(\boldsymbol{\omega}_t) \leq \mathcal{P}_{t+1}(\boldsymbol{\omega}_{t+1})$ and therefore the second requirement in Lemma 1 holds. Assuming that $\min_{\boldsymbol{\omega}} F(\boldsymbol{\omega}) = 0$, it is immediate to show that $\mathcal{P}_1(\boldsymbol{\omega}_1) = 0$ (first requirement). Finally, by definition we have that $\mathcal{P}_{m+1}(\boldsymbol{\omega}_{m+1}) = \mathcal{P}^{\star}$ and thus the third requirement holds as well. We have thus obtained a (hypothetical) mistake bound of the from given in (3). While this approach seems aesthetic, it is rather difficult to reason about the increase in the instantaneous primal objective functions due to the change in $\boldsymbol{\omega}$ and thus $\lambda$ might be excessively small and the bound is vacuous. In addition, we obtained the monotonicity property of the sequence $\mathcal{P}_1(\boldsymbol{\omega}_1), \ldots, \mathcal{P}_{m+1}(\boldsymbol{\omega}_{m+1})$ (second requirement in Lemma 1) by relying on the optimality of each $\boldsymbol{\omega}_t$ with respect to $\mathcal{P}_t(\boldsymbol{\omega})$. The optimality of $\boldsymbol{\omega}_t$ is a specific property of the FoReL algorithm and does not hold for many other online learning algorithms. These difficulties surface the alternative dual-based approach which we explore throughout this paper.

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 Boyd and Vandenberghe 2004). As we show in the sequel, the benefit in using the dual representation of $\mathcal{P}(\boldsymbol{\omega})$ is twofold. First, we are able to express the increase in the instantaneous dual representation of $\mathcal{P}(\boldsymbol{\omega})$ through a simple recursive update of the dual variables. Second, dual objective values are natural candidates for obtaining lower bounds for the optimal primal objective values. Thus, by switching to the dual representation we obtain a monotonically increasing sequence of *dual* objective values each of which is bounded above by $\mathcal{P}^{\star}$.

We now present an alternative view of the FoReL algorithm based on the notion of duality. This dual view would pave the way for analyzing online learning algorithms by setting $v_t$ in accordance to the instantaneous dual objective values. We formally show in Appendix 1 that the dual of the problem $\min_{\boldsymbol{\omega}} \mathcal{P}(\boldsymbol{\omega})$ is

$$\max_{\boldsymbol{\alpha} \in [0,C]^m} \mathcal{D}(\boldsymbol{\alpha}) \quad \text{where } \mathcal{D}(\boldsymbol{\alpha}) = \gamma \sum_{i=1}^{m} \alpha_i - G\left( \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \right). \tag{5}$$

The function $G$ is the Fenchel conjugate (Rockafellar 1970) of the function $F$ and is defined as follows,

$$G(\boldsymbol{\theta}) = \sup_{\boldsymbol{\omega} \in \Omega} \langle \boldsymbol{\omega}, \boldsymbol{\theta} \rangle - F(\boldsymbol{\omega}). \tag{6}$$

The weak duality theorem states that the maximum value of the dual problem is upper-bounded by the minimum value of the primal problem. Therefore, 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 $\mathcal{D}(\boldsymbol{\alpha}) \leq \mathcal{P}^{\star}$. Building on the definition of the instantaneous primal objective values, we denote by $\mathcal{D}_t$ the dual objective value of $\mathcal{P}_t$ which amounts to,

$$\mathcal{D}_t(\boldsymbol{\alpha}) = \gamma \sum_{i=1}^{t-1} \alpha_i - G\left( \sum_{i=1}^{t-1} \alpha_i y_i \mathbf{x}_i \right). \tag{7}$$

The instantaneous dual value $\mathcal{D}_t$ can also be cast as a mapping from $[0, C]^{t-1}$ into the reals. However, in contrast to the definition of the primal values, the instantaneous dual value $\mathcal{D}_t$ can be expressed as a specific assignment of the dual variables for the full dual

problem $\mathcal{D}$. Specifically, we obtain that for $(\alpha_1, \ldots, \alpha_{t-1}) \in [0, C]^{t-1}$ the following equality immediately holds,

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

Thus, the FoReL algorithm can alternatively be viewed as the process of finding a solution for the *dual* problem, $\max_{\boldsymbol{\alpha} \in [0,C]^m} \mathcal{D}(\boldsymbol{\alpha})$, where at the end of trial $t$ the online algorithm seeks a maximizer for the dual function confined to the first $t$ variables,

$$\max_{\boldsymbol{\alpha} \in [0,C]^m} \mathcal{D}(\boldsymbol{\alpha}) \quad \text{s.t.} \quad \forall i > t,\ \alpha_i = 0. \tag{8}$$

Analogous to our construction of instantaneous primal solutions, we construct a sequence of instantaneous assignments for the dual variables which we denote by $\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \ldots, \boldsymbol{\alpha}^{m+1}$ where $\boldsymbol{\alpha}^{t+1}$ is the maximizer of (8). The property of the dual objective that we utilize is that it can be optimized in a sequential manner. Namely, if on trial $t$ we ground $\alpha_i^t$ to zero for $i \geq t$ then $\mathcal{D}(\boldsymbol{\alpha}^t)$ does not depend on examples which have not been observed yet. Throughout the paper we assume that the supremum of $G(\boldsymbol{\theta})$ as defined in (6) is attainable. We show in Appendix 1, that the primal vector $\boldsymbol{\omega}_t$ can be derived from the dual vector $\boldsymbol{\alpha}^t$ through the equality,

$$\boldsymbol{\omega}_t = \operatorname*{argmax}_{\boldsymbol{\omega} \in \Omega} (\langle \boldsymbol{\omega}, \boldsymbol{\theta}_t \rangle - F(\boldsymbol{\omega})) \quad \text{where } \boldsymbol{\theta}_t = \sum_{i=1}^{m} \alpha_i^t y_i \mathbf{x}_i. \tag{9}$$

Furthermore, when $F(\boldsymbol{\omega})$ is convex, then strong duality holds and thus $\boldsymbol{\omega}_t$ as given in (9) is indeed the optimum of $\mathcal{P}_t(\boldsymbol{\omega})$ provided that $\boldsymbol{\alpha}^t$ is the optimum of (8).

We have thus presented two views of the FoReL algorithm through the prism of incremental optimization. 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 $\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^{m+1}$. The weak duality immediately enables us to cast an upper bound on the sequence of the corresponding dual values, $\forall t, \mathcal{D}(\boldsymbol{\alpha}^t) \leq \mathcal{P}^\star$, without resorting to or relying on optimality of any of the instantaneous dual solutions. Thus, by setting $v_t = \mathcal{D}(\boldsymbol{\alpha}^t)$ we immediately get that the third requirement from Lemma 1 holds. Next we show that the first requirement from Lemma 1 holds as well. 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 $\Omega$ is zero. Since $\boldsymbol{\alpha}^1$ is the zero vector we get that

$$v_1 = \mathcal{D}(\boldsymbol{\alpha}^1) = 0 - G(\mathbf{0}) = \inf_{\boldsymbol{\omega} \in \Omega} F(\boldsymbol{\omega}) = 0, \tag{10}$$

which implies that the first requirement from Lemma 1 hold. The monotonicity requirement from Lemma 1 follows directly from the fact that $\boldsymbol{\alpha}^{t+1}$ is the optimum of $\mathcal{D}(\boldsymbol{\alpha})$ over $[0, C]^t \times \{0\}^{m-t}$ while $\boldsymbol{\alpha}^t \in [0, C]^t \times \{0\}^{m-t}$.

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 (9). Naturally, we require that $\alpha_i^t = 0$ for all $i \geq t$ since otherwise $\boldsymbol{\omega}_t$ would depend on future examples which have not been observed yet. A key advantage of the dual representation is that we no longer need to find an optimal solution for each instantaneous dual problem $\mathcal{D}_t$. To prove that an online algorithm which operates on the dual variables entertains the mistake bound given in (3) it suffices to require that $\mathcal{D}(\boldsymbol{\alpha}^{t+1}) \geq \mathcal{D}(\boldsymbol{\alpha}^t)$. We show in the coming sections that few well studied algorithms can be analyzed using our primal-dual perspective. We do so by showing that the algorithms

INPUT: Complexity 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: $\hat{y}_t = \operatorname{sign}(\langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle)$

receive correct label $y_t$

find $\boldsymbol{\alpha}^{t+1} \in [0, C]^t \times \{0\}^{m-t}$ such that $\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t) \geq 0$

**Fig. 1** The template algorithm for online classification

guarantee a lower bound on the increase in the dual objective function on trials with prediction mistakes. Thus, all of the algorithms we analyze confine with the mistake bound given in (3) and differ in their choice of $F$ and in their mechanism for increasing the dual objective function.

To recap, we now describe a template algorithm for online classification which incrementally increases 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 $\boldsymbol{\omega}_t$ as given in (9). Next, we use $\boldsymbol{\omega}_t$ for predicting the label of $\mathbf{x}_t$, $\hat{y}_t = \operatorname{sign}(\langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle)$. Finally, in case of a prediction mistake we find a new dual solution $\boldsymbol{\alpha}^{t+1}$. This new dual solution is obtained by keeping the suffix of $m - t$ elements of $\boldsymbol{\alpha}^{t+1}$ at zero. The monotonicity requirement we imposed implies that the new value of the dual objective, $\mathcal{D}(\boldsymbol{\alpha}^{t+1})$, can only increase and cannot be smaller than $\mathcal{D}(\boldsymbol{\alpha}^t)$. Moreover, the average increase in the dual objective over erroneous trials should be strictly positive. In the next section we provide 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 by providing a general 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* $\min_{\boldsymbol{\omega} \in \Omega} F(\boldsymbol{\omega}) = 0$. *Let* $\mathcal{E} = \{t \in [m] : \hat{y}_t \neq y_t\}$ *and denote by* $\lambda$ *the* average *increase of the dual objective over the trials in* $\mathcal{E}$,

$$\lambda = \frac{1}{|\mathcal{E}|} \sum_{t \in \mathcal{E}} (\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t)).$$

*Then*,

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

*Proof* For all $t \in [m+1]$ define $v_t = \mathcal{D}(\boldsymbol{\alpha}^t)$. We prove the claim by applying Lemma 1 using the above assignments for the sequence $v_1, \ldots, v_{m+1}$. To do so, we need to show that the three requirements given in Lemma 1 hold. As in (10), the first requirement follows from the fact that $\boldsymbol{\alpha}^1 = \mathbf{0}$ and our assumption that $\min_{\boldsymbol{\omega} \in \Omega} F(\boldsymbol{\omega}) = 0$. The second requirement follows directly from the definition of the online algorithm in Fig. 1. Finally, the last requirement is a direct consequence of the weak duality theorem. □

The bound in Theorem 1 becomes useless when $\lambda$ 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 meaningful mistake bounds for each of the algorithm we discuss.

## 4 Analysis of quasi-additive online algorithms

In the previous section we introduced a general framework for online learning based on the notion of duality. In this section we analyze the family of quasi-additive online algorithms described in (Grove et al. 2001; Kivinen and Warmuth 1997, 2001) using the newly introduced dual view. This family includes several known algorithms such as the Perceptron algorithm (Rosenblatt 1958), Balanced-Winnow (Grove et al. 2001), and the family of $p$-norm algorithms (Gentile 2002).

Building on the exposition provided in the previous section we cast the online learning problem as the task of incrementally increasing the dual objective function given by (5). We show in this section that all quasi-additive online learning algorithms can be viewed as employing the same procedure for incrementing (5). The core difference between the algorithms we analyze distills to the complexity function $F$ which leads to different forms of the function $G$. We exploit this common ground 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 than prior proofs.

To guarantee an increase in the dual as given by (5) 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,

$$\alpha_i^{t+1} = \begin{cases} \alpha_i^t & \text{if } i \neq t, \\ C & \text{if } i = t. \end{cases} \tag{11}$$

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

In order to continue with the derivation and analysis of online algorithms, we now provide sufficient conditions for the update given by (11). The conditions guarantee an increase of the dual objective for all $t \in \mathcal{E}$ which is substantial enough to yield a mistake bound. 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,

$$\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t) = \gamma C - G(\boldsymbol{\theta}_t + C y_t \mathbf{x}_t) + G(\boldsymbol{\theta}_t), \tag{12}$$

where, to remind the reader, $\boldsymbol{\theta}_t = \sum_{i=1}^{t-1} \alpha_i^t y_i \mathbf{x}_i$. 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 Azoury and Warmuth 2001; Gentile 2002; Kivinen and Warmuth 1997, 2001).

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

$$G(\boldsymbol{\theta}_t + C y_t \mathbf{x}_t) = G(\boldsymbol{\theta}_t) + C y_t \langle \mathbf{x}_t, \boldsymbol{g}(\boldsymbol{\theta}_t) \rangle + \frac{1}{2} C^2 \langle \mathbf{x}_t, H(\boldsymbol{\theta}) \mathbf{x}_t \rangle. \tag{13}$$

Plugging the above equation into (12) gives that,

$$\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^{t}) = C(\gamma - y_t \langle \mathbf{x}_t, \boldsymbol{g}(\boldsymbol{\theta}_t) \rangle) - \frac{1}{2} C^2 \langle \mathbf{x}_t, H(\boldsymbol{\theta}) \mathbf{x}_t \rangle. \tag{14}$$

We next show that $\boldsymbol{\omega}_t = \boldsymbol{g}(\boldsymbol{\theta}_t)$ and therefore the second term in the right-hand of (13) 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,

$$G(\boldsymbol{\theta}_t) + \langle \boldsymbol{\omega}_t, \boldsymbol{\theta} - \boldsymbol{\theta}_t \rangle = \langle \boldsymbol{\omega}_t, \boldsymbol{\theta}_t \rangle - F(\boldsymbol{\omega}_t) + \langle \boldsymbol{\omega}_t, \boldsymbol{\theta} - \boldsymbol{\theta}_t \rangle = \langle \boldsymbol{\omega}_t, \boldsymbol{\theta} \rangle - F(\boldsymbol{\omega}_t). \tag{15}$$

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 (15) with the last inequality gives the following,

$$G(\boldsymbol{\theta}) \geq G(\boldsymbol{\theta}_t) + \langle \boldsymbol{\omega}_t, \boldsymbol{\theta} - \boldsymbol{\theta}_t \rangle. \tag{16}$$

Since (16) holds for all $\boldsymbol{\theta}$ it implies that $\boldsymbol{\omega}_t$ is a sub-gradient of $G$ at $\boldsymbol{\theta}_t$. 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 $\boldsymbol{\theta}_t$ 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 (14) we get that,

$$\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^{t}) = C(\gamma - y_t \langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle) - \frac{1}{2} C^2 \langle \mathbf{x}_t, H(\boldsymbol{\theta}) \mathbf{x}_t \rangle. \tag{17}$$

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 $\forall x \in \Omega : \langle \mathbf{x}, H(\boldsymbol{\theta}) \mathbf{x} \rangle \leq 1$ for all the particular choices of $G$ we analyze 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 Theorem 1 the update given by (11) ensures that,*

$$\lambda \geq \gamma C - \frac{1}{2} C^2.$$

We now provide concrete analyses for specific complexity functions $F$. For each choice of $F$ we derive the specific form the update given by (11) takes and briefly discuss the implications of the resulting mistake bounds.

*Example 1* (Perceptron) The Perceptron algorithm (Rosenblatt 1958) can be derived from (11) by setting $F(\boldsymbol{\omega}) = \frac{1}{2}\|\boldsymbol{\omega}\|^2$, $\Omega = \mathbb{R}^n$, and $\gamma = 1$. 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 $\boldsymbol{\omega}_t = \boldsymbol{\theta}_t$. The update $\boldsymbol{\omega}_{t+1} = \boldsymbol{g}(\boldsymbol{\theta}_{t+1})$ thus amounts to, $\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t + C y_t \mathbf{x}_t$, which is a *scaled* version of the well known Perceptron update. We now case the common assumption that the norm of all the instances is bounded and in particular we 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,

$$\left(C - \frac{1}{2}C^2\right) M \leq \min_{\boldsymbol{\omega} \in \mathbb{R}^n} \frac{1}{2}\|\boldsymbol{\omega}\|^2 + C \sum_{i=1}^{m} \ell(\boldsymbol{\omega}; (\mathbf{x}_i, y_i)). \tag{18}$$

On a first sight the above bound does not seem to take the form of one of the known mistake bounds for the Perceptron algorithm. We next show that since we are free to choose the constant $C$, which acts here as a simple scaling, we do obtain the tightest mistake bound that is known for the Perceptron. Note that on trial $t$, the hypothesis of the Perceptron can be rewritten as,

$$\boldsymbol{\omega}_t = C \sum_{i \in \mathcal{E}: i < t} y_i \mathbf{x}_i .$$

The above form implies that the predictions of the Perceptron algorithm do not depend on the actual value of $C$ so long as $C > 0$. Therefore, we can choose $C$ to be the minimizer of the bound given in (18) and rewrite the bound as,

$$\forall \boldsymbol{\omega} \in \mathbb{R}^n, M \leq \min_{C \in (0,2)} \left(\frac{1}{C(1 - \frac{1}{2}C)}\right) \left(\frac{1}{2}\|\boldsymbol{\omega}\|^2 + C \sum_{i=1}^{m} \ell(\boldsymbol{\omega}; (\mathbf{x}_i, y_i))\right), \tag{19}$$

where the domain $(0, 2)$ for $C$ ensures that the bound does not become vacuous. Finding the optimal value of $C$ for the right-hand side of the above and plugging this value back into the equation yields the following theorem.

**Theorem 2** *Let* $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ *be a sequence of examples such that* $\|\mathbf{x}_i\| \leq 1$ *for all* $i \in [m]$ *and assume that this sequence is presented to the Perceptron algorithm. Let* $\boldsymbol{\omega}$ *be an arbitrary vector in* $\mathbb{R}^n$ *and define* $L = \sum_{i=1}^{m} \ell(\boldsymbol{\omega}; (\mathbf{x}_i, y_i))$. *Then, the number of prediction mistakes of the Perceptron is upper bounded by,*

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

The proof of the theorem is given in Appendix 2. We would like to note that this bound is identical to the best known mistake bound for the Perceptron algorithm (see for example (Gentile 2002)). However, our proof technique is vastly different. Furthermore, the new technique also enables us to derive mistake and loss bounds for new algorithms such as the ones discussed in Sect. 5.

*Example 2* (Balanced Winnow) We now analyze a version of the Winnow algorithm called Balanced-Winnow (Grove et al. 2001) which is also closely related to the Exponentiated-Gradient algorithm (Kivinen and Warmuth 1997). For brevity we refer to the algorithm we analyze simply as Winnow. To derive the Winnow algorithm we choose,

$$F(\boldsymbol{\omega}) = \sum_{i=1}^{n} \omega_i \log\left(\frac{\omega_i}{1/n}\right), \tag{20}$$

and $\Omega = \Delta_n = \{\boldsymbol{\omega} \in \mathbb{R}^n_+ : \sum_{i=1}^{n} \omega_i = 1\}$. 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 Boyd and Vandenberghe 2004, p. 93),

$$G(\boldsymbol{\theta}) = \log\left(\frac{1}{n}\sum_{i=1}^{n}\exp(\theta_i)\right). \tag{21}$$

The $k$'th element of the gradient of $G$ is,

$$g_k(\boldsymbol{\theta}) = \frac{\exp(\theta_k)}{\sum_{i=1}^{n}\exp(\theta_i)}.$$

Note that $\boldsymbol{g}(\boldsymbol{\theta})$ is a vector in the $n$-dimensional probability simplex and therefore $\boldsymbol{\omega}_t = \boldsymbol{g}(\boldsymbol{\theta}_t) \in \Omega$. The $k$'th element of $\boldsymbol{\omega}_{t+1}$ can be rewritten using a multiplicative update rule,

$$\boldsymbol{\omega}_{t+1,k} = \frac{1}{Z_t}\exp(\theta_{t,k} + C y_t \mathbf{x}_{t,k}) = \frac{\boldsymbol{\omega}_{t,k}}{Z_t}\exp(C y_t \mathbf{x}_{t,k}), \tag{22}$$

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

To analyze the algorithm we need to show that $\langle \mathbf{x}_t, H(\boldsymbol{\theta})\mathbf{x}_t\rangle \leq 1$. The next lemma provides us with a general tool for bounding $\langle \mathbf{x}_t, H(\boldsymbol{\theta})\mathbf{x}_t\rangle$. The lemma gives conditions on $G$ which imply that its Hessian is diagonal dominant. A similar analysis of the Hessian was given in (Grove et al. 2001).

**Lemma 2** *Assume that $G(\boldsymbol{\theta})$ can be written as*,

$$G(\boldsymbol{\theta}) = \Psi\left(\sum_{r=1}^{n}\phi(\theta_r)\right),$$

*where $\phi$ and $\Psi$ are twice differentiable scalar functions. Denote by $\phi', \phi'', \Psi', \Psi''$ the first and second order derivatives of $\Psi$ and $\phi$. If $\Psi''(\sum_r \phi(\theta_r)) \leq 0$ for all $\boldsymbol{\theta}$ then*,

$$\langle \mathbf{x}, H(\boldsymbol{\theta})\mathbf{x}\rangle \leq \Psi'\left(\sum_{r=1}^{n}\phi(\theta_r)\right)\sum_{i=1}^{n}\phi''(\theta_i)x_i^2.$$

The proof of this lemma is given in Appendix 2.

We now rewrite $G(\boldsymbol{\theta})$ from (21) as $G(\boldsymbol{\theta}) = \Psi(\sum_{r=1}^{n}\phi(\theta_r))$ where $\Psi(s) = \log(s/n)$ and $\phi(\theta) = \exp(\theta)$. Note that $\Psi'(s) = 1/s$, $\Psi''(s) = -1/s^2$, and $\phi''(\theta) = \exp(\theta)$. We thus get that,

$$\Psi''\left(\sum_r \phi(\theta_r)\right) = -\left(\sum_r \exp(\theta_r)\right)^{-2} \leq 0.$$

Therefore, the conditions of Lemma 2 hold and we get that,

$$\langle \mathbf{x}, H(\boldsymbol{\theta})\mathbf{x}\rangle \leq \sum_{i=1}^{n}\frac{\exp(\theta_i)}{\sum_{r=1}^{n}\exp(\theta_r)}x_i^2 \leq \max_{i\in[n]} x_i^2.$$

Thus, if $\|\mathbf{x}_t\|_\infty \leq 1$ for all $t \in \mathcal{E}$ then we can apply corollary 1 and get the following mistake bound,

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

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

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

The bound above is typical of online algorithms which update their prediction mechanism in a multiplicative form as given by (22). The excessive loss suffered by the online algorithm above over the loss of any competitor scales logarithmically with the number of features.

*Example 3* ($p$-norm algorithms) We conclude this section with the analysis of the family of $p$-norm algorithms (Gentile 2002; Grove et al. 2001). This family can be viewed as a bridge between the Perceptron algorithm and the Winnow algorithm. As we show in the sequel, the Perceptron algorithm is a special case of a $p$-norm algorithm, obtained by setting $p = 2$, while the Winnow algorithm can be approximated by setting $p$ to a very large number. Formally, 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 Boyd and Vandenberghe 2004, p. 93) and the $i$'th element of the gradient of $G$ is,

$$g_i(\boldsymbol{\theta}) = \frac{\text{sign}(\theta_i)|\theta_i|^{p-1}}{\|\boldsymbol{\theta}\|_p^{p-2}}. \tag{23}$$

To analyze the $p$-norm algorithm we again use Lemma 2 and rewrite $G(\boldsymbol{\theta})$ as

$$G(\boldsymbol{\theta}) = \Psi\left(\sum_{r=1}^{n}\phi(\theta_r)\right),$$

where $\Psi(a) = \frac{1}{2}a^{2/p}$ and $\phi(a) = |a|^p$. Note that the first and second order derivatives are,

$$\Psi'(a) = \frac{1}{p}a^{2/p-1}, \qquad \Psi''(a) = \frac{1}{p}\left(\frac{2}{p} - 1\right)a^{2/p-2},$$
$$\phi''(a) = p(p-1)\text{sign}(a)|a|^{p-2}.$$

Therefore, if $p \geq 2$ then the conditions of Lemma 2 hold and we get that,

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

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 (24) gives,

$$\langle \mathbf{x}, H(\boldsymbol{\theta})\mathbf{x}\rangle \leq (p-1)\|\mathbf{x}\|_p^2.$$

If we impose the condition that $\|\mathbf{x}\|_p \leq \sqrt{1/(p-1)}$ then $\langle \mathbf{x}, H(\boldsymbol{\theta})\mathbf{x}\rangle \leq 1$. Recall that $\boldsymbol{\theta}_t$ for the update we employ can be written as,

$$\boldsymbol{\theta}_t = C \sum_{i\in\mathcal{E}:i<t} y_i \mathbf{x}_i.$$

Denote by $\mathbf{v} = \sum_{i\in\mathcal{E}:i<t} y_i\mathbf{x}_i$. Clearly, this vector does not depend on $C$. Since hypothesis $\boldsymbol{\omega}_t$ is defined from $\boldsymbol{\theta}_t$ as given by (23) we can rewrite the $j$'th component of $\boldsymbol{\omega}_t$ as,

$$C\frac{\text{sign}(v_j)|v_j|^{p-1}}{\|\mathbf{v}\|_p^{p-2}}.$$

Thus, similar to Example 1, the predictions of a $p$-norm algorithm which uses this update do not depend on the specific value of $C$ as long as $C > 0$. We now combine this fact with the assumption that $\|\mathbf{x}\|_p \leq \sqrt{1/(p-1)}$, and apply again corollary 1, to obtain that

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

As in the proof of Theorem 2, we can substitute $C$ with the minimizer of the above bound and obtain a general bound for the $p$-norm algorithm,

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

where as before $L = \sum_{i=1}^{m} \ell(\boldsymbol{\omega}; (\mathbf{x}_i, y_i))$.

## 5 Deriving and analyzing new online learning algorithms

In the previous section we described the family of quasi-additive online learning algorithms. The algorithms are based on the simple update procedure defined in (11) which leads to a conservative increase of the dual objective since we modify a *single* variable of $\boldsymbol{\alpha}$ by setting it to a *constant* value. Furthermore, such an update takes place solely on trials for which there was a prediction mistake ($t \in \mathcal{E}$). The purpose of this section is two fold. First, we describe a broader and, in practice, more powerful update procedures which, based on the actual predictions, may modify multiple elements of $\boldsymbol{\alpha}$. Second, we provide an alternative analysis in the form of regret bounds, rather than mistake bounds. The motivation for the new algorithms is as follows. Intuitively, update schemes which yield larger increases of the dual objective value on each online trial are likely to "consume" more of the upper bound

on the total possible increase in the dual as set by $\mathcal{P}^\star$. Thus, they are in practice likely to suffer smaller number of mistakes. Moreover, setting the dual variables in accordance to the loss that is suffered on each trial allows us to derive bounds on the cumulative *loss* of the online algorithms rather than merely bounding the number of *mistakes* the algorithms make. We start this section with a very brief overview of the regret model in which the loss of the online algorithm is compared to the loss of any fixed competitor. We then describe a few new online update procedures and analyze them in the regret model.

The mistake bounds presented thus far are inherently deficient as they provide a bound on the *number* of mistakes through the *hinge-loss* of the competitor. In contrast, *regret* bounds measure the performance of the online algorithm and the competitor using the same loss function. The regret of an online algorithm compared to a fix predictor, denoted $\boldsymbol{\omega}$, is defined to be the following difference,

$$\frac{1}{m}\sum_{i=1}^{m}\ell^{\gamma}(\boldsymbol{\omega}_i;(\mathbf{x}_i,y_i)) - \frac{1}{m}\sum_{i=1}^{m}\ell^{\gamma}(\boldsymbol{\omega};(\mathbf{x}_i,y_i)).$$

The right-hand summand in the above expression reflects the loss that is suffered by using a fix predictor $\boldsymbol{\omega}$ for all $i \in [m]$. In particular, the vector $\boldsymbol{\omega}$ can be set in hindsight to be the vector which minimizes the cumulative loss on the observed sequence of $m$ instances. Naturally, the problem of finding the vector $\boldsymbol{\omega}$ which minimizes the right-hand summand above depends on the entire sequence of examples. The regret thus reflects the amount of excess loss suffered by the online algorithm due lack of knowledge of the entire sequence. In this paper we derive regret bounds which are tailored to the hinge-loss function. The bounds follow again our primal-dual perspective which incorporates a complexity term for $\boldsymbol{\omega}$ through a function $F : \Omega \to \mathbb{R}$. The regret bound we present in this section takes the form,

$$\forall \boldsymbol{\omega} \in \Omega, \quad \frac{1}{m}\sum_{i=1}^{m}\ell^{\gamma}(\boldsymbol{\omega}_i;(\mathbf{x}_i,y_i)) - \frac{1}{m}\sum_{i=1}^{m}\ell^{\gamma}(\boldsymbol{\omega};(\mathbf{x}_i,y_i)) \leq \sqrt{\frac{2F(\boldsymbol{\omega})}{m}}. \tag{25}$$

Thus, this bound implies that the regret of the online algorithm with respect to any vector whose complexity grows slower than $m$ approaches zero as $m$ goes to infinity.

### 5.1 Aggressive quasi-additive online algorithms

The update scheme we described in Sect. 4 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 $\alpha_t^{t+1}$ to $C$ guarantees a sufficient increase in the dual, there might be other values $\alpha_t^{t+1}$ which would lead to even 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 $\alpha_i^{t+1}$ adaptively, depending on the loss suffered on trial $t$. This improved update constructs $\boldsymbol{\alpha}^{t+1}$ as follows,

$$\alpha_i^{t+1} = \begin{cases} \alpha_i^t & \text{if } i \neq t, \\ \min\{\ell^{\gamma}(\boldsymbol{\omega}_t;(\mathbf{x}_t,y_t)), C\} & \text{if } i = t. \end{cases} \tag{26}$$

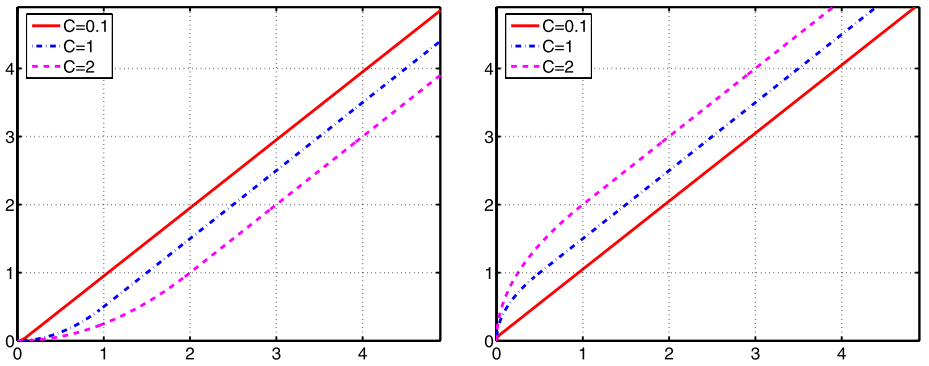


**Fig. 2** The mitigating function $\mu(x)$ (*left*) and its inverse (*right*) for different values of $C$

In contrast to the previous update which modified $\boldsymbol{\alpha}$ only when there was a prediction mistake, the new update modifies $\boldsymbol{\alpha}$ whenever $\ell^{\gamma}(\boldsymbol{\omega}_t; (\mathbf{x}_t, y_t)) > 0$. As before, the above update can be used with various complexity functions for $F$, yielding different aggressive quasi-additive algorithms. 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 $\gamma$.

We now provide a unified analysis for all algorithms which are based on the update given by (26). To do so we define the following function,

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

The function $\mu(\cdot)$ is invertible on $\mathbb{R}_+$ and we denote its inverse function by $\mu^{-1}(\cdot)$. A straightforward calculation gives that

$$\mu^{-1}(x) = \begin{cases} x + \frac{1}{2}C & \text{if } x \geq \frac{1}{2}C, \\ \sqrt{2Cx} & \text{otherwise.} \end{cases}$$

The functions $\mu(\cdot)$ and $\mu^{-1}(\cdot)$ are illustrated in Fig. 2. Applying $\mu$ to losses smaller than $C$ lessens the extent of the loss. Therefore, we also refer to $\mu$ as a mitigating function. Note, though, that $\mu(\cdot)$ and $\mu^{-1}(\cdot)$ become very similar to the identity function for small values of $C$. The following theorem provides a bound on the cumulative sum of $\ell^{\gamma}(\boldsymbol{\omega}_t, (\mathbf{x}_t, y_t))$.

**Theorem 3** *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 which satisfies* $\min_{\boldsymbol{\omega} \in \Omega} F(\boldsymbol{\omega}) = 0$. *Assume we run an online algorithm whose update is based on* (26) *while 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,*

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

*Proof* We first show that

$$\sum_{t=1}^{m} \mu(\ell^{\gamma}(\boldsymbol{\omega}_t; (\mathbf{x}_t, y_t))) \leq \sum_{t=1}^{m} \ell^{\gamma}(\boldsymbol{\omega}; (\mathbf{x}_t, y_t)) + \frac{F(\boldsymbol{\omega})}{C}, \tag{27}$$

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 $\mathcal{D}(\boldsymbol{\alpha}^{m+1}) = \sum_{t=1}^{m} (\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t))$. Define $\tau_t = \min\{\ell^{\gamma}(\boldsymbol{\omega}_t; (\mathbf{x}_t, y_t)), C\}$ and note that the sole difference between the updates given by (26) and (11) is that $\tau_t$ replaces $C$. Thus, the derivation of (17) in Sect. 4 can be repeated almost verbatim with $\tau_t$ replacing $C$ to obtain that,

$$\mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t) \geq \tau_t(\gamma - y_t \langle \boldsymbol{\omega}_t, \mathbf{x}_t \rangle) - \frac{1}{2}\tau_t^2. \tag{28}$$

Summing over $t \in [m]$, rewriting $\tau_t$ as the minimum between $C$ and the loss at time $t$, and rearranging terms while using the definition of $\mu(\cdot)$, we get that,

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

Comparing the lower and upper bounds on $\mathcal{D}(\boldsymbol{\alpha}^{m+1})$ and rearranging terms yield the inequality provided in (27). We now divide (27) by $m$ and use the fact that $\mu$ is convex to get that

$$\begin{aligned} \mu\left(\frac{1}{m}\sum_{t=1}^{m} \ell^{\gamma}(\boldsymbol{\omega}_t; (\mathbf{x}_t, y_t))\right) &\leq \frac{1}{m}\sum_{t=1}^{m} \mu(\ell^{\gamma}(\boldsymbol{\omega}_t; (\mathbf{x}_t, y_t))) \\ &\leq \frac{1}{m}\sum_{t=1}^{m} \ell^{\gamma}(\boldsymbol{\omega}; (\mathbf{x}_t, y_t)) + \frac{F(\boldsymbol{\omega})}{mC}. \end{aligned} \tag{29}$$

Finally, since both sides of the above inequality are non-negative and since $\mu^{-1}$ is a monotonically increasing function we can apply $\mu^{-1}$ to both sides of (29) to get the bound stated in the theorem. □

While the bound stated in the above theorem is no longer in the form of a mistake bound, it nonetheless does not provide a regret bound of the form given by (25). We now show that the bound of Theorem 3 can indeed be distilled and cast in the form of a loss bound, similar to (25), by choosing appropriately the parameter $C$. To do so, we note that $\mu^{-1}(x) \leq x + \frac{1}{2}C$. Therefore, the right-hand side of the bound in Theorem 3 is bounded above by

$$\frac{1}{m}\sum_{t=1}^{m} \ell^{\gamma}(\boldsymbol{\omega}; (\mathbf{x}_t, y_t)) + \frac{F(\boldsymbol{\omega})}{Cm} + \frac{1}{2}C. \tag{30}$$

Note that $C$ both divides the complexity function $F(\boldsymbol{\omega})$ as well as appears as an independent term. Choosing $C$ such that the terms $\frac{F(\boldsymbol{\omega})}{Cm}$ and $\frac{1}{2}C$ yields the tightest loss bound for this update, we obtain the following corollary.

**Corollary 2** *Assume we run an online algorithm whose update is based on* (26) *under the same conditions stated in Theorem* 3 *while choosing*

$$C = \sqrt{\frac{2F(\boldsymbol{\omega})}{m}},$$

*then*,

$$\frac{1}{m}\sum_{t=1}^{m} \ell^{\gamma}(\boldsymbol{\omega}_t; (\mathbf{x}_t, y_t)) - \frac{1}{m}\sum_{t=1}^{m} \ell^{\gamma}(\boldsymbol{\omega}; (\mathbf{x}_t, y_t)) \leq \sqrt{\frac{2F(\boldsymbol{\omega})}{m}}.$$

We can also derive a mistake bound from (29). To do so, we note that $\ell^{\gamma}(\boldsymbol{\omega}_t; (\mathbf{x}_t, y_t)) \geq \gamma$ whenever the algorithm makes a prediction mistake. Since $\mu$ is a monotonically increasing function and since $\ell^{\gamma}(\cdot)$ is a non-negative function, we get that

$$\sum_{t\in\mathcal{E}} \mu(\gamma) \leq \sum_{t=1}^{m} \mu(\ell^{\gamma}(\boldsymbol{\omega}_t; (\mathbf{x}_t, y_t))) \leq \frac{F(\boldsymbol{\omega})}{C} + \sum_{t=1}^{m} \ell^{\gamma}(\boldsymbol{\omega}; (\mathbf{x}_t, y_t)).$$

Thus, we obtain the mistake bound,

$$M \leq \frac{\mathcal{P}^{\star}}{\lambda} \quad \text{where } \lambda \geq C\mu(\gamma) = \begin{cases} \gamma C - \frac{1}{2}C^2 & \text{if } C \leq \gamma, \\ \frac{1}{2}\gamma^2 & \text{if } C > \gamma. \end{cases} \tag{31}$$

Our focus thus far was on an update which modifies a *single* dual variable, albeit aggressively. We now examine another implication of our analysis which suggests the modification of *multiple* dual variables on each trial. A simple argument presented below implies that this broader family of updates also achieves the mistake and regret bounds above.

### 5.2 Updating multiple dual variables

The new update given in (26) is advantageous over the previous conservative update given in (11) 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 the modification of a single dual variable on each trial. We nonetheless can increase the dual more dramatically by modifying multiple dual variables on each trial. We now outline two forms of updates which modify multiple dual variables on each trial.

In the first update scheme we optimize the dual over a set of dual variables $I_t \subseteq [t]$ which includes $t$. Given $I_t$, we set $\boldsymbol{\alpha}^{t+1}$ to be,

$$\boldsymbol{\alpha}^{t+1} = \operatorname*{argmax}_{\boldsymbol{\alpha}\in[0,C]^m} \mathcal{D}(\boldsymbol{\alpha}) \quad \text{s.t.} \quad \forall i \notin I_t,\ \alpha_i = \alpha_i^t. \tag{32}$$

This more general update also achieves the bound of Theorem 3 and the minimal increase in the dual as given by (31). To see this, note that the requirement that $t \in I_t$ implies,

$$\mathcal{D}(\boldsymbol{\alpha}^{t+1}) \geq \max\{\mathcal{D}(\boldsymbol{\alpha}) : \boldsymbol{\alpha} \in [0, C]^m \text{ and } \forall i \neq t, \alpha_i = \alpha_i^t\}. \tag{33}$$

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 examine a few choices for $I_t$. Setting $I_t = [t]$ for all $t$ gives the FoReL algorithm we mentioned in Sect. 3. 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 optimization procedure. In contrast, (32) 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 $\alpha_t^{t+1}$ 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 (Crammer et al. 2005) 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 (Crammer et al. 2005). 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 (32) analytically while gaining modest increases in the dual. The full power of the update is unleashed for large values of $k$. However, (32) cannot be solved analytically and requires the usage of numerical QP solvers based on, for instance, interior point methods.

The second update scheme modifies multiple dual variables on each trial as well, alas it does not require solving an optimization problem with multiple variables. Instead, we perform $k_t$ mini-updates each of which focuses on a single variable from the set $[t]$. Formally, let $i_1, \ldots, i_{k_t}$ be a sequence of indices such that $i_1 = t$ and $i_j \in [t]$ for all $j \in [k_t]$. We define a sequence of dual solutions in a recursive manner as follows. We start by setting $\hat{\boldsymbol{\alpha}}^0 = \boldsymbol{\alpha}^t$ and then perform a sequence of single variable updates of the form,

$$\hat{\boldsymbol{\alpha}}^j = \operatorname*{argmax}_{\boldsymbol{\alpha} \in [0,C]^m} \mathcal{D}(\boldsymbol{\alpha}) \quad \text{s.t.} \quad \forall p \neq i_j, \hat{\alpha}_p^j = \hat{\alpha}_p^{j-1}.$$

Finally, we update $\boldsymbol{\alpha}^{t+1} = \hat{\boldsymbol{\alpha}}^{k_t}$. In words, we first decide on an ordering of the dual variables that defined $\boldsymbol{\omega}_t$ and incrementally increase the dual by fixing all the dual variables but the current one that is considered. For this variable we find the optimal solution of the constrained dual. The first dual variable we update is $\alpha_t$ thus ensuring that the first step in the row of updates is identical to the Passive Aggressive update which was mentioned above. Indeed, note that for $k_t = 1$ this update is identical to the update given in (32) with $I_t = \{t\}$. Since at each operation we can only increase the dual we immediately conclude that Theorem 3 holds for this composite update scheme as well. The main advantage of this update is its simplicity since each operation involves optimization over a single variable which can be solved analytically. The increase in the dual due to this update is closely related to the so called row action methods in optimization (see for example Censor and Zenios 1997).

## 6 On the connection to previous analyses

The main contribution of this paper is the introduction of a framework for the design and analysis of online prediction algorithms. There exist though voluminous amounts of work that employ different approaches for the analysis of online algorithms. In this section, we draw a few connections to earlier analysis techniques by modifying the primal problem defined on the right hand side of (2). Our modifications naturally lead to modified dual problems. We then analyze the increase in the modified duals to draw connections to prior work and analyses.

To remind the reader, in order to obtain a mistake bound of the from given in (3) we associated a quality value, $v_t$, with each weight vector $\boldsymbol{\omega}_t$. We then analyzed the progress

of the online algorithm by monitoring the difference $\Delta_t \stackrel{\text{def}}{=} v_{t+1} - v_t$. Our quality values are based on the dual objective values of the primal problem,

$$\min_{\boldsymbol{\omega}} \mathcal{P}(\boldsymbol{\omega}) \quad \text{where } \mathcal{P}(\boldsymbol{\omega}) = F(\boldsymbol{\omega}) + C \sum_{i=1}^{m} (\gamma - y_i \langle \boldsymbol{\omega}, \mathbf{x}_i \rangle)_+ .$$

Concretely, we set $v_t = \mathcal{D}(\boldsymbol{\alpha}^t)$ and use the increase in the dual as our notion of progress. Furthermore, the mistake and regret bounds above were derived by reasoning about the increase in the dual due to prediction mistakes.

Most if not all previous work analyzed online algorithms by measuring the quality of $\boldsymbol{\omega}_t$ based on the correlation or distance between $\boldsymbol{\omega}_t$ and a fixed (yet unknown to the online algorithm) competitor, denoted here by $\mathbf{u}$. For example, Novikoff's analysis of the Perceptron (Novikoff 1962) is based on the inner product between $\mathbf{u}$ and the current prediction $\boldsymbol{\omega}_t$, $v_t = \langle \boldsymbol{\omega}_t, \mathbf{u} \rangle$. Another quality measure, which has been vastly used in previous analyses of online algorithms, is based on the squared Euclidean distance, $v_t = \|\boldsymbol{\omega}_t - \mathbf{u}\|^2$ (see for example Azoury and Warmuth 2001; Gentile 2002; Kivinen and Warmuth 1997, 2001 and the references therein). We show in the sequel that we can represent these previous definitions of $v_t$ as an instantaneous value of a dual objective by modifying the primal problem.

The first simple modification of the primal problem that we present replaces the single margin parameter $\gamma$ with trial dependent parameters $\gamma_1, \ldots, \gamma_m$. Each trial dependent margin parameter, $\gamma_i$, is set in accordance to example $i$ and the fixed competitor $\mathbf{u}$. Formally, let $\mathbf{u}$ be a fixed competitor and set $\gamma_i = y_i \langle \mathbf{u}, \mathbf{x}_i \rangle$. We now define the loss on trial $t$ to be the hinge-loss for a target margin value of $\gamma_t$. With this modification on hand we obtain the following primal problem,

$$\begin{aligned} \mathcal{P}(\boldsymbol{\omega}) &= F(\boldsymbol{\omega}) + C \sum_{i=1}^{m} (\gamma_i - y_i \langle \boldsymbol{\omega}, \mathbf{x}_i \rangle)_+ \\ &= F(\boldsymbol{\omega}) + C \sum_{i=1}^{m} (y_i \langle \mathbf{u}, \mathbf{x}_i \rangle - y_i \langle \boldsymbol{\omega}, \mathbf{x}_i \rangle)_+ . \end{aligned}$$

By construction, the loss suffered by $\mathbf{u}$ on each trial $i$ is zero since the margin $\mathbf{u}$ attains is exactly $\gamma_i$. Thus, the primal objective attained by $\mathbf{u}$ consists solely of the complexity term of $\mathbf{u}$, $F(\mathbf{u})$. Since $\mathcal{P}(\mathbf{u})$ upper bounds the optimal value of the primal we get that,

$$\min_{\boldsymbol{\omega}} \mathcal{P}(\boldsymbol{\omega}) \leq \mathcal{P}(\mathbf{u}) = F(\mathbf{u}).$$

Moving to the dual of this newly introduced primal problem, we get that the dual of the aforementioned primal problem is

$$\mathcal{D}(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \gamma_i \alpha_i - G(\boldsymbol{\theta}) \quad \text{where } \boldsymbol{\theta} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i .$$

Note that the mere difference between the above dual form and the dual of the original problem as described by (5) distills to replacing the fixed margin value $\gamma$ with a trial dependent one $\gamma_i$. Since $\gamma_i = y_i \langle \mathbf{u}, \mathbf{x}_i \rangle$, we can further rewrite the dual as follows,

$$\mathcal{D}(\boldsymbol{\alpha}) = \left\langle \mathbf{u}, \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \right\rangle - G(\boldsymbol{\theta}) = \langle \mathbf{u}, \boldsymbol{\theta} \rangle - G(\boldsymbol{\theta}). \tag{34}$$

We now embark on a specific connection to prior work by examining the case where $F(\boldsymbol{\omega}) = \frac{1}{2}\|\boldsymbol{\omega}\|^2$. For this choice of $F$, the Fenchel conjugate $G$ amounts to $G(\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{\theta}\|^2$ and we get that the dual further simplifies to the following form,

$$\mathcal{D}(\boldsymbol{\alpha}) = \langle \mathbf{u}, \boldsymbol{\theta} \rangle - \frac{1}{2}\|\boldsymbol{\theta}\|^2 = -\frac{1}{2}\|\boldsymbol{\theta} - \mathbf{u}\|^2 + \frac{1}{2}\|\mathbf{u}\|^2.$$

The change in the value of the dual objective due to a change in the dual variables from $\boldsymbol{\alpha}^t$ to $\boldsymbol{\alpha}^{t+1}$ amounts to,

$$\Delta_t = \mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t) = \frac{1}{2}(\|\boldsymbol{\theta}_t - \mathbf{u}\|^2 - \|\boldsymbol{\theta}_{t+1} - \mathbf{u}\|^2).$$

Furthermore, the specific choice of $F$ implies that $\boldsymbol{\omega}_t = \boldsymbol{\theta}_t$ (see also the analysis of the Perceptron algorithm in Sect. 4). Thus, the change in the dual can be written solely in terms of the primal vectors $\boldsymbol{\omega}_t$, $\boldsymbol{\omega}_{t+1}$ and the competitor $\mathbf{u}$,

$$\Delta_t = \frac{1}{2}(\|\boldsymbol{\omega}_t - \mathbf{u}\|^2 - \|\boldsymbol{\omega}_{t+1} - \mathbf{u}\|^2).$$

We thus ended up with the notion of progress which corresponds to the quality measure $v_t = \|\boldsymbol{\omega}_t - \mathbf{u}\|^2$.

Before proceeding to deriving the next quality measure from our framework, we would like to underscore the fact that our primal-dual perspective readily leads to a mistake bound for this choice of primal problem. Concretely, since $\min_{\boldsymbol{\omega}\in\Omega} \frac{1}{2}\|\boldsymbol{\omega}\|^2 = 0$, the initial vector $\boldsymbol{\omega}_1$, which is obtained by setting all the dual variables $\alpha_i^1$ to zero, corresponds to a dual objective function whose value is zero. Combining the form of the increase in the dual with the fact that the minimum of the primal is bounded above by $F(\mathbf{u}) = \frac{1}{2}\|\mathbf{u}\|^2$ we get that,

$$\sum_{t=1}^{m}(\|\boldsymbol{\omega}_t - \mathbf{u}\|^2 - \|\boldsymbol{\omega}_{t+1} - \mathbf{u}\|^2) \leq \|\mathbf{u}\|^2. \tag{35}$$

If we now use the Perceptron's update, $\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t + C y_t \mathbf{x}_t$ we get that the left hand side of (35) further upper bounds the following expression,

$$\sum_{t\in\mathcal{E}}(2C y_t \langle \mathbf{u}, \mathbf{x}_t \rangle - C^2\|\mathbf{x}_t\|^2). \tag{36}$$

As in the original mistake bound proof of the Perceptron, let us assume that the norm of the competitor $\mathbf{u}$ is 1 and that it classifies the entire sequence correctly with a margin of at least $\gamma$. Thus $y_t\langle \mathbf{u}, \mathbf{x}_t \rangle \geq \gamma$ for all $t$. Assume in addition that all the instances reside in a ball of radius $R$ we get that (36) is bounded below by

$$M(2C\gamma - C^2R^2) = MC(2\gamma - CR^2).$$

Choosing $C = \gamma/R^2$ and recalling (35) we obtain the well known mistake bound of the Perceptron,

$$M\frac{\gamma}{R^2}\left(2\gamma - \frac{\gamma}{R^2}R^2\right) \leq \|\mathbf{u}\|^2 = 1 \quad \Rightarrow \quad M \leq \left(\frac{R}{\gamma}\right)^2.$$

To recap, we have shown that a simple modification of the primal problem leads to a notion of progress that amounts to the change in the distance between the competitor and the *primal*

vector that is used for prediction. We also illustrated that our framework can be used again to derive a mistake bound by casting a simple bound on the primal objective function, and bounding from below the increase in the dual.

Next, we show that Novikoff's measure of quality, $v_t = \langle \boldsymbol{\omega}_t, \mathbf{u} \rangle$, employed in the analysis of the Perceptron (Novikoff 1962) can be obtained from our framework by a different choice of $F$. Our starting point is again the choice of trial-dependent hinge-loss which resulted the following bound,

$$\sum_{t=1}^{m} \Delta_t \leq F(\mathbf{u}). \tag{37}$$

Next, note that for the purpose of our analysis we are free to choose the complexity function $F$ in hindsight. In particular, we use the predictors constructed by the online algorithm in the definition of $F$. Let us defer the specific form of $F$ and initially define it in the following, rather abstract, form, $F(\boldsymbol{\omega}) = U\|\boldsymbol{\omega}\|$. In addition, we keep using the trial-dependent margin losses. The dual objective thus again takes the form given by (34), namely, $D(\boldsymbol{\alpha}) = \langle \mathbf{u}, \boldsymbol{\theta} \rangle - G(\boldsymbol{\theta})$. The Fenchel conjugate of the 2-norm is a barrier function (see again (Boyd and Vandenberghe 2004)). Concretely, for our choice of $F$ we get that its Fenchel conjugate is,

$$G(\boldsymbol{\theta}) = \begin{cases} 0 & \|\boldsymbol{\theta}\| \leq U, \\ \infty & \text{otherwise.} \end{cases}$$

Therefore, we get that $D(\boldsymbol{\alpha}) = \langle \boldsymbol{\theta}, \mathbf{u} \rangle$ so long as $\boldsymbol{\theta}$ is inside the ball of radius $U$ and otherwise $D(\boldsymbol{\alpha}) = -\infty$. In addition, let us choose $\boldsymbol{\omega}_t = \boldsymbol{\theta}_t$ for all $t \in [T]$. (Note that here we do not use the definition of $\boldsymbol{\omega}_t$ as in (9). Nevertheless, our general primal-dual framework does not rely on this particular choice.) To ensure that $G(\boldsymbol{\theta}_t)$ is finite we now define $U$ to be $\max_{t \in [T]} \|\boldsymbol{\omega}_t\|$ and thus $\mathcal{D}(\boldsymbol{\alpha}^t) = \langle \boldsymbol{\omega}_t, \mathbf{u} \rangle$ for all $t \in [T]$. These specific choices of $F$ and $U$ imply that the increase in the dual objective takes the following simple form,

$$\Delta_t = \mathcal{D}(\boldsymbol{\alpha}^{t+1}) - \mathcal{D}(\boldsymbol{\alpha}^t) = \langle \boldsymbol{\omega}_{t+1}, \mathbf{u} \rangle - \langle \boldsymbol{\omega}_t, \mathbf{u} \rangle.$$

The reader familiar with the original mistake bound proof of the Perceptron would immediately recognize the above term as the measure of progress used by the proof. Indeed, plugging the Perceptron update in the above equation we get that on trials with a prediction mistake $\Delta_t$ is,

$$\Delta_t = \langle \boldsymbol{\omega}_t + y_t \mathbf{x}_t, \mathbf{u} \rangle - \langle \boldsymbol{\omega}_t, \mathbf{u} \rangle = y_t \langle \mathbf{x}_t, \mathbf{u} \rangle.$$

On the rest of the trials there is no change in the dual objective and thus $\Delta_t = 0$. We now assume, as in the original mistake bound proof of the Perceptron algorithm, that the norm of the competitor $\mathbf{u}$ is 1 and that it classifies the entire sequence correctly with a margin of at least $\gamma$. The second assumption translates to the classical lower bound,

$$\sum_{t=1}^{m} \Delta_t = \sum_{t \in \mathcal{E}} y_t \langle \mathbf{u}, \mathbf{x}_t \rangle \geq M\gamma.$$

From the mistake bound proof of the Perceptron we know that the norm of $\boldsymbol{\omega}_t$ (which equals $\boldsymbol{\theta}_t$) is at most $\sqrt{M}R$ where $R$ is the radius of the ball encapsulating all of the examples. We therefore get the following upper bound on the primal objective,

$$\mathcal{P}(\mathbf{u}) = F(\mathbf{u}) = \left( \max_t \|\boldsymbol{\omega}_t\| \right) \|\mathbf{u}\| \leq \sqrt{M}R.$$

We now tie the lower bound on $\sum_t \Delta_t$ with its upper bound using (37) to get that,

$$M\gamma \leq \sum_{t=1}^{m} \Delta_t \leq F(\mathbf{u}) \leq \sqrt{M}R \quad \Rightarrow \quad \sqrt{M} \leq \frac{R}{\gamma},$$

which after squaring yields the celebrated Perceptron's mistake bound.

We have thus shown that two well studied quality measures and their corresponding notions of progress can be derived and analyzed using the primal-dual paradigm suggested in this paper. The core difference in the two analyses amounts to two different choices of the complexity function $F$. We conclude this section by drawing a connection between online methods that construct their prediction as a sequence of instantaneous optimization problems and our framework. We start by reviewing the notion of Bregman divergences.

A Bregman divergence (Bregman 1967) is defined via a strictly convex function $F : \Omega \to \mathbb{R}$ defined on a closed, convex set $\Omega \subseteq \mathbb{R}^n$. A Bregman function $F$ needs to satisfy a set of constraints. We omit the description of the specific constraints and refer the reader to (Censor and Zenios 1997). The Bregman divergence is derived through the function $F$ as follows,

$$B_F(\boldsymbol{\omega}||\mathbf{u}) = F(\boldsymbol{\omega}) - (F(\mathbf{u}) + \langle \nabla F(\mathbf{u}), (\boldsymbol{\omega} - \mathbf{u}) \rangle).$$

That is, $B_F$ measures the difference between $F$ at $\boldsymbol{\omega}$ and its first-order Taylor expansion about $\mathbf{u}$, evaluated again at $\boldsymbol{\omega}$. Bregman divergences generalize some commonly studied distance and divergence measures.

Kivinen and Warmuth (1997) provided a general scheme for online learning. In their scheme the predictor $\boldsymbol{\omega}_{t+1}$ constructed at the end of trial $t$ from the current prediction $\boldsymbol{\omega}_t$ is defined as the solution to the following problem,

$$\boldsymbol{\omega}_{t+1} = \operatorname*{argmin}_{\boldsymbol{\omega} \in \Omega} B_F(\boldsymbol{\omega}||\boldsymbol{\omega}_t) + C\ell(\boldsymbol{\omega}; (\mathbf{x}_t, y_t)). \tag{38}$$

That is, the new predictor should maintain a small Bregman divergence to the current predictor while attaining a small loss. The constant $C$ mitigates between these two, typically conflicting, requirements. We now show that when the loss function is the hinge-loss, the problem defined by (38) can be viewed as a special case of our framework. For the hinge-loss we can rewrite (38) as follows,

$$\min_{\boldsymbol{\omega} \in \Omega, \xi_t \in \mathbb{R}_+} B_F(\boldsymbol{\omega}||\boldsymbol{\omega}_t) + C\xi_t \quad \text{s.t.} \quad y_t \langle \boldsymbol{\omega}, \mathbf{x}_t \rangle \geq \gamma - \xi_t.$$

In Appendix 1 we show that the dual of the above problem is the following problem,

$$\max_{\eta_t \in [0,C]} \gamma\eta_t - G(\boldsymbol{\theta}_t + \eta_t y_t \mathbf{x}_t).$$

Furthermore, $\boldsymbol{\theta}_t$ satisfies the following recursive form,

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \eta_t y_t \mathbf{x}_t.$$

An examination of the above dual problem immediately reveals that this dual problem can be obtained from the dual problem defined in (34) by setting $\alpha_i = \eta_i$ for $i \leq t$ and $\alpha_i = 0$ for $i > t$. Therefore, the problem defined by Kivinen and Warmuth can be viewed as a special case of one of the schemes discussed in Sect. 5.2. Concretely, we update only the variable $\alpha_t^t$ by setting it to $\eta_t$ and leave the rest of the dual variables intact, in particular $\alpha_i^t = \alpha_i^{t+1} = 0$ for all $i > t$.

## 7 Discussion

We presented a new framework for the design and analysis of online learning algorithms. Our framework yields the tightest known bounds for quasi-additive online classification algorithms. The new framework also paves the way to new algorithms. There are various possible extensions of the work that we plan to pursue. Our framework can be naturally 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 other losses such as the logistic loss. We also plan to explore 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 Dekel et al. 2005).

**Acknowledgements** Thanks to the anonymous reviewers for helpful comments. This work was supported by the Israeli Science Foundation, grant No. 039-7444.

## Appendix 1 Derivations of the dual problems

In this section we derive the dual problems of the main primal problems introduced in this paper. We start with the dual of the minimization problem $\min_{\boldsymbol{\omega}\in\Omega} \mathcal{P}(\boldsymbol{\omega})$ where

$$\mathcal{P}(\boldsymbol{\omega}) = F(\boldsymbol{\omega}) + C \sum_{i=1}^{m} \ell^{\gamma}(\boldsymbol{\omega}; (\mathbf{x}_i, y_i)). \tag{39}$$

Using the definition of $\ell^{\gamma}$ we can rewrite the optimization problem as,

$$\inf_{\boldsymbol{\omega}\in\Omega, \boldsymbol{\xi}\in\mathbb{R}^m_+} F(\boldsymbol{\omega}) + C \sum_{i=1}^{m} \xi_i \quad \text{s.t.} \quad \forall i \in [m], \quad y_i \langle \boldsymbol{\omega}, \mathbf{x}_i \rangle \geq \gamma - \xi_i . \tag{40}$$

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

$$\inf_{\boldsymbol{\omega}\in\Omega, \boldsymbol{\xi}\in\mathbb{R}^m_+} \sup_{\boldsymbol{\alpha}\in\mathbb{R}^m_+} \underbrace{F(\boldsymbol{\omega}) + C \sum_{i=1}^{m} \xi_i + \sum_{i=1}^{m} \alpha_i (\gamma - y_i \langle \boldsymbol{\omega}, \mathbf{x}_i \rangle - \xi_i)}_{\stackrel{\text{def}}{=} \mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\xi}, \boldsymbol{\alpha})} . \tag{41}$$

Equation (41) is equivalent to (40) 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 $\alpha_i$ in (41) is zero. If on the other hand the constraint does not hold then $\alpha_i$ equals $\infty$, which implies that $\boldsymbol{\omega}$ cannot constitute the optimal primal solution. The dual objective function is defined to be,

$$\mathcal{D}(\boldsymbol{\alpha}) = \inf_{\boldsymbol{\omega}\in\Omega, \boldsymbol{\xi}\in\mathbb{R}^m_+} \mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\xi}, \boldsymbol{\alpha}). \tag{42}$$

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

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

The last term is equal 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,

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

The second term in the above presentation of $\mathcal{D}(\boldsymbol{\alpha})$ can be rewritten as $G(\sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i)$ where $G$ is the Fenchel conjugate[1] of $F(\boldsymbol{\omega})$, as given in (6). Thus, for $\boldsymbol{\alpha} \in [0, C]^m$ the dual objective function can be written as,

$$\mathcal{D}(\boldsymbol{\alpha}) = \gamma \sum_{i=1}^{m} \alpha_i - G\left( \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \right). \tag{43}$$

Next, we derive the dual of the problem introduced at the end of Sect. 6. To remind the reader, the primal problem is,

$$\min_{\boldsymbol{\omega}\in\Omega, \xi_t\in\mathbb{R}_+} B_F(\boldsymbol{\omega}||\boldsymbol{\omega}_t) + C\xi_t \quad \text{s.t.} \quad y_t \langle \boldsymbol{\omega}, \mathbf{x}_t \rangle \geq \gamma - \xi_t. \tag{44}$$

Following the same line of derivation used for obtaining the dual of the previous problem, we form the Lagrangian and separate it into terms, each of which depends only on a subset of the problem variables. Denoting the Lagrange multiplier for the single constraint in (44) by $\eta_t$, we obtain the following,

$$D(\eta_t) = \gamma \eta_t - \sup_{\boldsymbol{\omega}\in\Omega} (\langle \boldsymbol{\omega}, \eta_t y_t \mathbf{x}_t \rangle - B_F(\boldsymbol{\omega}||\boldsymbol{\omega}_t)),$$

where $\eta_t$ should reside in $[0, C]$. We now write explicitly the Bregman divergence term and omit constants to obtain the more direct form,

$$D(\eta_t) = \gamma \eta_t - \sup_{\boldsymbol{\omega}\in\Omega} (\langle \boldsymbol{\omega}, \eta_t y_t \mathbf{x}_t \rangle - F(\boldsymbol{\omega}) + \langle \nabla F(\boldsymbol{\omega}_t), \boldsymbol{\omega} \rangle).$$

The gradient of $F$, $\nabla F$, is typically denoted by $f$. The mapping defined by $f$ is the inverse of the link function $g$ introduced in Sect. 4 (see also the list of references pointed to at that section). We thus denote by $\boldsymbol{\theta}_t$ the image of $\boldsymbol{\omega}_t$ under $f$, $\boldsymbol{\theta}_t = \nabla F(\boldsymbol{\omega}_t) = f(\boldsymbol{\omega}_t)$. Equipped with this notation we can rewrite $D(\eta_t)$ as follows,

$$D(\eta_t) = \gamma \eta_t - \sup_{\boldsymbol{\omega}\in\Omega} (\langle \boldsymbol{\omega}, \boldsymbol{\theta}_t + \eta_t y_t \mathbf{x}_t \rangle - F(\boldsymbol{\omega})).$$

Using $G$ again to denote the Fenchel conjugate of $F$ we get that the dual of the problem defined in (44) is,

$$D(\eta_t) = \gamma \eta_t - G(\boldsymbol{\theta}_t + \eta_t y_t \mathbf{x}_t). \tag{45}$$

[1] In cases where $F$ is differentiable with an invertible gradient, $G$ is also called the Legendre transform of $F$. See for example (Boyd and Vandenberghe 2004).

Let us denote by $\boldsymbol{\omega}_{t+1}$ the optimum of the primal problem. Since $F$ is twice differentiable, it is immediate to verify that the vector $\boldsymbol{\omega}_{t+1}$ must satisfy the following condition,

$$f(\boldsymbol{\omega}_{t+1}) = f(\boldsymbol{\omega}_t) + \eta_t y_t \mathbf{x}_t \Rightarrow \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \eta_t y_t \mathbf{x}_t. \tag{46}$$

## Appendix 2 Technical proofs

*Proof of Theorem 2* First note that if $L = 0$ then the setting $C = 1$ in (19) yields the bound $M \leq \|\boldsymbol{\omega}\|^2$ which is identical to the bound stated by the theorem for the case $L = 0$. We thus focus on the case $L > 0$ and we prove the theorem by finding the value of $C$ which minimizes the right-hand side of (19) for $C$. To simplify our notation we define $B = L/\|\boldsymbol{\omega}\|^2$ and denote,

$$\rho(C) = \frac{1}{(1 - \frac{1}{2}C)}\left(\frac{1}{2C}\|\boldsymbol{\omega}\|^2 + L\right) = \frac{\|\boldsymbol{\omega}\|^2}{(1 - \frac{1}{2}C)}\left(\frac{1}{2C} + B\right). \tag{47}$$

The function $\rho(C)$ is convex in $C$ and to find its minimum we can simply take its derivative with respect to $C$ and find the zero of the derivative. The derivative of $\rho$ with respect to $C$ is,

$$\rho'(C) = \frac{\|\boldsymbol{\omega}\|^2}{2(1 - \frac{1}{2}C)^2}\left(B - \frac{1 - C}{C^2}\right).$$

Comparing $\rho'(C)$ to zero while omitting multiplicative constants gives the following quadratic equation,

$$BC^2 + C - 1 = 0.$$

The larger root of the above equation is,

$$\begin{aligned} C &= \frac{\sqrt{1+4B} - 1}{2B} = \left(\frac{\sqrt{1+4B} - 1}{2B}\right)\left(\frac{\sqrt{1+4B} + 1}{\sqrt{1+4B} + 1}\right) \\ &= \frac{4B}{2B(\sqrt{1+4B} + 1)} = \frac{2}{\sqrt{1+4B} + 1}. \end{aligned} \tag{48}$$

It is easy to verify that the above value of $C$ is always in $(0, 2)$ and therefore it is the minimizer of $\rho(C)$ over $(0, 2)$. Plugging (48) into (47) and rearranging terms gives,

$$\begin{aligned} \rho(C) &= \|\boldsymbol{\omega}\|^2\left(\frac{1}{1 - \frac{1}{\sqrt{1+4B}+1}}\right)\left(\frac{\sqrt{1+4B}+1}{4} + B\right) \\ &= \frac{\|\boldsymbol{\omega}\|^2}{4}\left(\frac{\sqrt{1+4B}+1}{\sqrt{1+4B}}\right)(\sqrt{1+4B} + (1+4B)) \\ &= \frac{\|\boldsymbol{\omega}\|^2}{4}(\sqrt{1+4B}+1)^2 = \frac{\|\boldsymbol{\omega}\|^2}{4}(2 + 4B + 2\sqrt{1+4B}). \end{aligned}$$

Finally, the definition of $B$ implies that,

$$\rho(C) = L + \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \frac{1}{2}\sqrt{\|\boldsymbol{\omega}\|^4 + 4L\|\boldsymbol{\omega}\|^2}.$$

This concludes our proof. □

*Proof of Lemma 2* Using the chain rule we get that,

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

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,

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

where the last inequality follows from the assumption that $\Psi''(\sum_r \phi(\theta_r)) \leq 0$. This concludes our proof. □

## References


Azoury, K., & Warmuth, M. (2001). Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, *43*(3), 211–246.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge University Press.

Bregman, L. M. (1967). The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, *7*, 200–217.

Censor, Y., & Zenios, S. A. (1997). *Parallel optimization: theory, algorithms, and applications*. New York: Oxford University Press.

Cesa-Bianchi, N., Conconi, A., & Gentile, C. (2002). On the generalization ability of on-line learning algorithms. In *Advances in neural information processing systems* (Vol. 14, pp. 359–366).

Cesa-Bianchi, N., Conconi, A., & Gentile, C. (2005). A second-order perceptron algorithm. *SIAM Journal on Computing*, *34*(3), 640–668.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2005). *Online passive aggressive algorithms*. Technical report, The Hebrew University.

Dekel, O., Shalev-Shwartz, S., & Singer, Y. (2005). The forgetron: a kernel-based perceptron on a fixed budget. In *Advances in neural information processing systems* (Vol. 18).

Gentile, C. (2001). A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research*, *2*, 213–242.

Gentile, C. (2002). The robustness of the p-norm algorithms. *Machine Learning*, *53*(3).

Grove, A. J., Littlestone, N., & Schuurmans, D. (2001). General convergence results for linear discriminant updates. *Machine Learning*, *43*(3), 173–210.

Hannan, J. (1957). Approximation to Bayes risk in repeated play. In M. Dresher, A. W. Tucker, & P. Wolfe (Eds.), *Contributions to the theory of games* (Vol. III, pp. 97–139). Princeton: Princeton University Press.

Helmbold, D. P., Kivinen, J., & Warmuth, M. (1999). Relative loss bounds for single neurons. *IEEE Transactions on Neural Networks*, *10*(6), 1291–1304.

Kivinen, J., & Warmuth, M. (1997). Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, *132*(1), 1–64.

Kivinen, J., & Warmuth, M. (2001). Relative loss bounds for multidimensional regression problems. *Journal of Machine Learning*, *45*(3), 301–329.

Kivinen, J., Smola, A. J., & Williamson, R. C. (2002). Online learning with kernels. *IEEE Transactions on Signal Processing*, *52*(8), 2165–2176.

Li, Y., & Long, P. M. (2002). The relaxed online maximum margin algorithm. *Machine Learning*, *46*(1–3), 361–387.

Littlestone, N. (1988). Learning when irrelevant attributes abound: a new linear-threshold algorithm. *Machine Learning*, *2*, 285–318.

Littlestone, N. (1989). *Mistake bounds and logarithmic linear-threshold learning algorithms*. PhD thesis, U.C. Santa Cruz, March 1989.

Novikoff, A. B. J. (1962). On convergence proofs on perceptrons. In *Proceedings of the symposium on the mathematical theory of automata* (Vol. XII, pp. 615–622).

Rockafellar, R. T. (1970). *Convex analysis*. Princeton: Princeton University Press.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*, 386–407. (Reprinted in *Neurocomputing*, MIT Press, 1988.)

Vovk, V. (2001). Competitive on-line statistics. *International Statistical Review*, *69*, 213–248.