Online Multitask Learning

Ofer Dekel[1], Philip M. Long[2], and Yoram Singer[1,2]

[1] School of Computer Science and Engineering, The Hebrew University, Jerusalem 91904, Israel
[2] Google, 1600 Amphitheater Parkway, Mountain View, CA 94043, USA
{oferd,singer}@cs.huji.ac.il plong@google.com

Abstract. We study the problem of online learning of multiple tasks in parallel. On each online round, the algorithm receives an instance and makes a prediction for each one of the parallel tasks. We consider the case where these tasks all contribute toward a common goal. We capture the relationship between the tasks by using a single global loss function to evaluate the quality of the multiple predictions made on each round. Specifically, each individual prediction is associated with its own individual loss, and then these loss values are combined using a global loss function. We present several families of online algorithms which can use any absolute norm as a global loss function. We prove worst-case relative loss bounds for all of our algorithms.

1 Introduction

Multitask learning is the problem of learning several related problems in parallel. In this paper, we discuss the multitask learning problem in the online learning context. We focus on the possibility that the learning tasks contribute toward a common goal. Our hope is that we can benefit by taking account of this to learn the tasks jointly, as opposed to learning each task independently.

For concreteness, we focus on the task of binary classification, and note that our algorithms and analysis can be adapted to regression problems using ideas in [1]. In the online multitask classification setting, we are faced with k separate online binary classification problems, in parallel. The online learning process takes place in a sequence of rounds. At the beginning of round t, the algorithm observes a set of k instances, one for each of the binary classification problems. The algorithm predicts the binary label of each of the instances it has observed, and then receives the correct label of each instance. At this point, each of the algorithm's predictions is associated with a non-negative loss, and we use $\boldsymbol{\ell}_t = (\ell_{t,1}, \ldots, \ell_{t,k})$ to denote the k-coordinate vector whose elements are the individual loss values of the respective tasks. Assume that we selected, ahead of time, a *global loss* function $\mathcal{L} : \mathbb{R}^k \to \mathbb{R}_+$, which is used to combine these individual loss values into a single number, and define the global loss attained on round t to be $\mathcal{L}(\boldsymbol{\ell}_t)$. At the end of the online round, the algorithm may use the k new labeled examples it has obtained to improve its prediction mechanism for the rounds to come. The goal of the learning algorithm is to suffer the smallest possible cumulative loss over the course of T rounds, $\sum_{t=1}^{T} \mathcal{L}(\boldsymbol{\ell}_t)$.

The choice of the global loss function captures the overall consequences of the individual prediction errors, and therefore how the algorithm prioritizes correcting errors. For example, if $\mathcal{L}(\boldsymbol{\ell}_t)$ is defined to be $\sum_{j=1}^{k} \ell_{t,j}$ then the online algorithm is penalized equally for errors on any of the tasks; this results in effectively treating the tasks independently. On the other hand, if $\mathcal{L}(\boldsymbol{\ell}_t) = \max_j \ell_{t,j}$ then the algorithm is only interested in the worst mistake made on every round. We do not assume that the datasets of the various tasks are similar or otherwise related. Moreover, the examples presented to the algorithm for each task may come from different domains and may possess different characteristics. The multiple tasks are tied together by the way we define the objective of our algorithm.

In this paper, we focus on the case where the global loss function is an *absolute norm*. A norm $\|\cdot\|$ is a function such that $\|\mathbf{v}\| > 0$ for all $\mathbf{v} \neq 0$, $\|0\| = 0$, $\|\lambda\mathbf{v}\| = |\lambda|\|\mathbf{v}\|$ for all $\mathbf{v}$ and all $\lambda \in \mathbb{R}$, and which satisfies the triangle inequality. A norm is said to be absolute if $\|\mathbf{v}\| = \||\mathbf{v}|\|$ for all $\mathbf{v}$, where $|\mathbf{v}|$ is obtained by replacing each component of $\mathbf{v}$ with its absolute value. The most well-known family of absolute norms is the family of p-norms (also called L_p norms), defined for all $p \geq 1$ by $\|\mathbf{v}\|_p = (\sum_{j=1}^{n} |v_j|^p)^{1/p}$. A special member of this family is the L_∞ norm, which is defined to be the limit of the above when p tends to infinity, and can be shown to equal $\max_j |v_j|$. A less popular family of absolute norms is the family of r-max norms. For any integer r between 1 and k, the r-max norm of $\mathbf{v} \in \mathbb{R}^k$ is the sum of the absolute values of the r absolutely largest components of $\mathbf{v}$. Formally,

$$\|\mathbf{v}\|_{r\text{-max}} \;=\; \sum_{j=1}^{r} |v_{\pi(j)}| \;\; \text{where} \;\; |v_{\pi(1)}| \geq |v_{\pi(2)}| \geq \;\; \ldots \;\; \geq |v_{\pi(k)}| \;\; .$$

Note that both the L_1 norm and L_∞ norm are special cases of the r-max norm, as well as being p-norms.

On each online round, we balance a trade-off between retaining the information acquired on previous rounds and modifying our hypotheses based on the new examples obtained on this round. Instead of balancing this trade-off individually for each of the learning tasks, as would be done naively, we balance it for all of the tasks jointly. By doing so, we allow ourselves to make bigger modifications to some of the hypotheses at the expense of the others. To motivate our approach, we present a handful of concrete examples.

Multiclass Classification using the L_∞ Norm Assume that we are faced with a multiclass classification problem, where the size of the label set is k. One way of solving this problem is by learning k binary classifiers, where each classifier is trained to distinguish between one of the classes and the rest of the classes, the *one-vs-rest* method. If all of the binary classifiers make correct predictions, we can correctly predict the multiclass label. Otherwise, a single binary mistake is as bad as many binary mistakes. Therefore, we only care about the worst binary prediction on round t, and we do so by setting the global loss to be $\|\boldsymbol{\ell}_t\|_\infty$.

Vector-Valued Regression using the L_2 Norm Let us deviate momentarily from the binary classification setting, and assume that we are faced with multiple

regression problems. Specifically, assume that our task is to predict the three-dimensional position of an object. Each of the three coordinates is predicted using an individual regressor, and the regression loss for each task is simply the absolute difference between the true and the predicted value on the respective axis. In this case, the most appropriate global loss function is the L_2 norm, which maps the vector of individual losses to the Euclidean distance between the true and predicted 3-D targets. (Note that we take the actual Euclidean distance and not the squared Euclidean distance often minimized in regression settings).

Error Correcting Output Codes and the r-max *Norm* Error Correcting Output Codes is a technique for reducing a multiclass classification problem to multiple binary classification problems [2]. The power of this technique lies in the fact that a correct multiclass prediction can be made even when a few of the binary predictions are wrong. The reduction is represented by a code matrix $M \in \{-1,+1\}^{s,k}$, where s is the number of multiclass labels and k is the number of binary problems used to encode the original multiclass problem. Each row in M represents one of the s multiclass labels, and each column induces one of the k binary classification problems. Given a multiclass training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$, with labels $y_i \in \{1,\ldots,s\}$, the binary problem induced by column j is to distinguish between the positive examples $\{(\mathbf{x}_i, y_i : M_{y_i,j} = +1\}$ and negative examples $\{(\mathbf{x}_i, y_i : M_{y_i,j} = -1\}$. When a new instance is observed, applying the k binary classifiers to it gives a vector of binary predictions, $\hat{\mathbf{y}} = (\hat{y}_1,\ldots,\hat{y}_k) \in \{-1,+1\}^k$. We then predict the multiclass label of this instance to be the index of the row in M which is closest to $\hat{\mathbf{y}}$ in Hamming distance. Define the *code distance* of M, denoted by $d(M)$, to be the minimal Hamming distance between any two rows in M. It is straightforward to show that a correct multiclass prediction can be guaranteed as long as the number of binary mistakes made on this instance is less than $d(M)/2$. In other words, making $d(M)/2$ binary mistakes is as bad as making more binary mistakes. Let $r = d(M)/2$. If the binary classifiers are trained in the online multitask setting, we should only be interested in whether the r'th largest loss is less than 1, which would imply that a correct multiclass prediction can be guaranteed. Regretfully, taking the r'th largest element of a vector (in absolute value) does not constitute a norm and thus does not fit in our setting. However, the r-max norm defined above can serve as a proxy.

In this paper, we present three families of online multitask algorithms. Each family includes algorithms for all the absolute norms. All of the algorithms presented in this paper follow the general skeleton outlined in Fig. 1. Specifically, all of our algorithms use an additive update rule, which enables us to transform them into kernel methods. For each algorithm we prove a relative loss bound, namely, we show that the cumulative global loss attained by the algorithm is not much greater than the cumulative loss attained by any fixed set of k linear hypotheses, even one defined in hindsight.

Much previous work on theoretical and applied multitask learning has concerned how to take advantage of cases in which a number of learning problems are related [3–8]; in contrast, we do not assume that the tasks are related and instead we consider how to take account of common consequences of errors. Kivinen and

input: norm $\|\cdot\|$

initialize: $\mathbf{w}_{1,1} = \ldots = \mathbf{w}_{1,k} = (0, \ldots, 0)$

for $t = 1, 2, \ldots$

- receive $\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,k}$
- predict $\operatorname{sign}(\mathbf{w}_{t,j} \cdot \mathbf{x}_{t,j})$ $[1 \le j \le k]$
- receive $y_{t,1}, \ldots, y_{t,k}$
- calculate $\ell_{t,j} = \left[1 - y_{t,j}\mathbf{w}_{t,j} \cdot \mathbf{x}_{t,j}\right]_+$ $[1 \le j \le k]$
- suffer loss $\ell_t = \|(\ell_{t,1}, \ldots, \ell_{t,n})\|$
- update $\mathbf{w}_{t+1,j} = \mathbf{w}_{t,j} + \tau_{t,j} y_{t,j} \mathbf{x}_{t,j}$ $[1 \le j \le k]$

Fig. 1. A general skeleton for an online multitask classification algorithm. A concrete algorithm is obtained by specifying the values of $\tau_{t,j}$.

Warmuth [9] generalized the notion of matching loss [10] to multi-dimensional outputs; this enables analysis of algorithms that perform multi-dimensional regression by composing linear functions with a variety of transfer functions. It is not obvious how to directly use their work to address the problem of linear classification with dependent losses addressed in this paper. An analysis of the L_∞ norm of prediction errors is implicit in some past work of Crammer and Singer [11, 12]; the present paper extends this work to a broader framework, and tightens the analysis. When k, the number of multiple tasks, is set to 1, two of the algorithms presented in this paper reduce to the PA-I algorithm [1].

This paper is organized as follows. In Sec. 2 we present our problem more formally and prove a key lemma which facilitates the analysis of our algorithms. In Sec. 3 we present our first family of algorithms, which works in the finite horizon online setting. In Sec. 4 we extend the first family of algorithms to the infinite horizon setting. Finally, in Sec. 5, we present our third family of algorithms for the multitask setting, and show that it shares the analyses of both previous algorithms. The third family of algorithms requires solving a small optimization problem on each online round. Finally, in Sec. 6, we discuss some efficient techniques for solving this optimization problem.

2 Online Multitask Learning with Additive Updates

We begin by presenting the online multitask setting more formally. We are faced with k online binary classification problems in parallel. The instances of each problem are drawn from separate instance domains, and for concreteness, we assume that the instances of problem j are all vectors in $\mathbb{R}^{n_j}$. As stated in the previous section, online learning is performed in a sequence of rounds. On round t, the algorithm observes k instances, $(\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,k}) \in \mathbb{R}^{n_1} \times \ldots \times \mathbb{R}^{n_k}$. The algorithm maintains k separate classifiers in its internal memory, one for

each of the multiple tasks, and updates them from round to round. Each of these classifiers is a margin-based linear predictor, defined by a weight vector. Let $\mathbf{w}_{t,j} \in \mathbb{R}^{n_j}$ denote the weight vector used to define the j'th linear classifier on round t. The algorithm uses its classifiers to predict the binary labels $\hat{y}_{t,1}, \ldots, \hat{y}_{t,k}$, where $\hat{y}_{t,j} = \text{sign}(\mathbf{w}_{t,j} \cdot \mathbf{x}_{t,j})$. Then, the correct labels of the respective problems, $y_{t,1}, \ldots, y_{t,k}$, are revealed and each one of the predictions is evaluated. We use the hinge-loss function to penalize incorrect predictions, namely, the loss associated with the j'th problem is defined to be

$$\ell_{t,j} \;=\; \left[1 - y_{t,j}\mathbf{w}_{t,j} \cdot \mathbf{x}_{t,j}\right]_+ \;\;,$$

where $[a]_+ = \max\{0, a\}$. As previously stated, the global loss is then defined to be $\|\boldsymbol{\ell}_t\|$, where $\|\cdot\|$ is a predefined absolute norm. Finally, the algorithm applies an update to each of the online hypotheses, and defines the vectors $\mathbf{w}_{t+1,1}, \ldots, \mathbf{w}_{t+1,k}$. All of the algorithms presented in this paper use an additive update rule, and define $\mathbf{w}_{t+1,j}$ to be $\mathbf{w}_{t,j} + \tau_{t,j} y_{t,j} \mathbf{x}_{t,j}$, where $\tau_{t,j}$ is a non-negative scalar. The algorithms only differ from one another in the way they set $\tau_{t,j}$. The general skeleton followed by all of our online algorithms is given in Fig. 1.

A concept of key importance in this paper is the notion of the *dual norm* [13]. Any norm $\|\cdot\|$ defined on $\mathbb{R}^n$ has a dual norm, also defined on $\mathbb{R}^n$, denoted by $\|\cdot\|^\star$ and given by

$$\|\mathbf{u}\|^\star \;=\; \max_{\mathbf{v} \in \mathbb{R}^n} \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{v}\|} \;=\; \max_{\mathbf{v} \in \mathbb{R}^n \,:\, \|\mathbf{v}\| = 1} \mathbf{u} \cdot \mathbf{v} \;\;. \tag{1}$$

The dual of a p-norm is itself a p-norm, and specifically, the dual of $\|\cdot\|_p$ is $\|\cdot\|_q$, where $\frac{1}{q} + \frac{1}{p} = 1$. The dual of $\|\cdot\|_\infty$ is $\|\cdot\|_1$ and vice versa. It can also be shown that the dual of $\|\mathbf{v}\|_{r\text{-max}}$ is

$$\|\mathbf{u}\|^\star_{r\text{-max}} \;=\; \max\left\{\|\mathbf{u}\|_\infty, \frac{\|\mathbf{u}\|_1}{r}\right\} \;\;. \tag{2}$$

An important property of dual norms, which is an immediate consequence of Eq. (1), is that for any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ it holds that

$$\mathbf{u} \cdot \mathbf{v} \;\leq\; \|\mathbf{u}\|^\star \, \|\mathbf{v}\| \;\;. \tag{3}$$

If $\|\cdot\|$ is a p-norm then the above is known as Hölder's inequality, and specifically if $p = 2$ then it is called the Cauchy-Schwartz inequality. Two additional properties which we rely on are that the dual of the dual norm is the original norm (see for instance [13]), and that the dual of an absolute norm is also an absolute norm. As previously mentioned, to obtain concrete online algorithms, all that remains is to define the update weights $\tau_{t,j}$. The different ways of setting $\tau_{t,j}$ discussed in this paper all share the following properties:

- *boundedness:* $\forall\, 1 \leq t \leq T \quad \|\boldsymbol{\tau}_t\|^\star \leq C$ for some predefined parameter C
- *non-negativity:* $\forall\, 1 \leq t \leq T,\ 1 \leq j \leq k \quad \tau_{t,j} \geq 0$
- *conservativeness:* $\forall\, 1 \leq t \leq T,\ 1 \leq j \leq k \quad (\ell_{t,j} = 0) \;\Rightarrow\; (\tau_{t,j} = 0)$

Even before specifying the exact value of $\tau_{t,j}$, we can state and prove a powerful lemma which is the crux of our analysis. This lemma will motivate and justify our specific choices of $\tau_{t,j}$ throughout this paper.

Lemma 1. *Let $\{(\mathbf{x}_{t,j}, y_{t,j})\}_{1\le t\le T}^{1\le j\le k}$ be a sequence of T k-tuples of examples, where each $\mathbf{x}_{t,j} \in \mathbb{R}^{n_j}$, $\|\mathbf{x}_{t,j}\|_2 \le R$ and each $y_{t,j} \in \{-1,+1\}$. Let $\mathbf{w}_1^\star, \ldots, \mathbf{w}_k^\star$ be arbitrary vectors where $\mathbf{w}_j^\star \in \mathbb{R}^{n_j}$, and define the hinge loss attained by $\mathbf{w}_j^\star$ on example $(\mathbf{x}_{t,j}, y_{t,j})$ to be $\ell_{t,j}^\star = \left[1 - y_{t,j}\mathbf{w}_j^\star \cdot \mathbf{x}_{t,j}\right]_+$. Let $\|\cdot\|$ be a norm and let $\|\cdot\|^\star$ denote its dual. Assume we apply an algorithm of the form outlined in Fig. 1 to this sequence, where the update satisfies the boundedness property with $C > 0$, as well as the non-negativity and conservativeness properties. Then*

$$\sum_{t=1}^{T}\sum_{j=1}^{k}\Big(2\tau_{t,j}\ell_{t,j} - \tau_{t,j}^2\|\mathbf{x}_{t,j}\|_2^2\Big) \;\le\; \sum_{j=1}^{k}\|\mathbf{w}_j^\star\|_2^2 \;+\; 2C\sum_{t=1}^{T}\|\boldsymbol{\ell}_t^\star\| \quad .$$

Proof. Define $\Delta_{t,j} = \|\mathbf{w}_{t,j} - \mathbf{w}_j^\star\|_2^2 - \|\mathbf{w}_{t+1,j} - \mathbf{w}_j^\star\|_2^2$. We prove the lemma by bounding $\sum_{t=1}^{T}\sum_{j=1}^{k}\Delta_{t,j}$ from above and from below. Beginning with the upper bound, we note that for each $1 \le j \le k$, $\sum_{t=1}^{T}\Delta_{t,j}$ is a telescopic sum which collapses to

$$\sum_{t=1}^{T}\Delta_{t,j} \;=\; \|\mathbf{w}_{1,j} - \mathbf{w}^\star\|_2^2 - \|\mathbf{w}_{T+1,j} - \mathbf{w}^\star\|_2^2 \quad .$$

Using the facts that $\mathbf{w}_{1,j} = (0,\ldots,0)$ and $\|\mathbf{w}_{T+1,j} - \mathbf{w}^\star\|_2^2 \ge 0$ for all $1 \le j \le k$, we conclude that

$$\sum_{t=1}^{T}\sum_{j=1}^{k}\Delta_{t,j} \;\le\; \sum_{j=1}^{k}\|\mathbf{w}_j^\star\|_2^2 \quad . \tag{4}$$

Turning to the lower bound, we note that we can consider only non-zero summands which actually contribute to the sum, namely $\Delta_{t,j} \neq 0$. Plugging the definition of $\mathbf{w}_{t+1,j}$ into $\Delta_{t,j}$, we get

$$\begin{aligned}
\Delta_{t,j} &= \|\mathbf{w}_{t,j} - \mathbf{w}_j^\star\|_2^2 - \|\mathbf{w}_{t,j} + \tau_{t,j}y_{t,j}\mathbf{x}_{t,j} - \mathbf{w}_j^\star\|_2^2 \\
&= \tau_{t,j}\left(-2y_{t,j}\mathbf{w}_{t,j}\cdot\mathbf{x}_{t,j} - \tau_{t,j}\|\mathbf{x}_{t,j}\|_2^2 + 2y_{t,j}\mathbf{w}_j^\star\cdot\mathbf{x}_{t,j}\right) \\
&= \tau_{t,j}\left(2(1 - y_{t,j}\mathbf{w}_{t,j}\cdot\mathbf{x}_{t,j}) - \tau_{t,j}\|\mathbf{x}_{t,j}\|_2^2 - 2(1 - y_{t,j}\mathbf{w}_j^\star\cdot\mathbf{x}_{t,j})\right) \quad .
\end{aligned} \tag{5}$$

Since our update is conservative, $\Delta_{t,j} \neq 0$ implies that $\ell_{t,j} = 1 - y_{t,j}\mathbf{w}_{t,j}\cdot\mathbf{x}_{t,j}$. By definition, it also holds that $\ell_{t,j}^\star \ge 1 - y_{t,j}\mathbf{w}_j^\star\cdot\mathbf{x}_{t,j}$. Using these two facts in Eq. (5) and using the fact that $\tau_{t,j} \ge 0$ gives $\Delta_{t,j} \ge \tau_{t,j}(2\ell_{t,j} - \tau_{t,j}\|\mathbf{x}_{t,j}\|_2^2 - 2\ell_{t,j}^\star)$. Summing this inequality over $1 \le j \le k$ gives

$$\sum_{j=1}^{k}\Delta_{t,j} \;\ge\; \sum_{j=1}^{k}\left(2\tau_{t,j}\ell_{t,j} - \tau_{t,j}^2\|\mathbf{x}_{t,j}\|_2^2\right) - 2\sum_{j=1}^{k}\tau_{t,j}\ell_{t,j}^\star \quad . \tag{6}$$

Using Eq. (3) we know that $\sum_{j=1}^{k} \tau_{t,j}\ell^\star_{t,j} \leq \|\boldsymbol{\tau}_t\|^\star \|\boldsymbol{\ell}^\star_t\|$. Using our assumption that $\|\boldsymbol{\tau}_t\|^\star \leq C$, we have that $\sum_{j=1}^{k} \tau_{t,j}\ell^\star_{t,j} \leq C\|\boldsymbol{\ell}^\star_t\|$. Plugging this inequality into Eq. (6) gives

$$\sum_{j=1}^{k} \Delta_{t,j} \;\geq\; \sum_{j=1}^{k} \left(2\tau_{t,j}\ell_{t,j} - \tau^2_{t,j}\|\mathbf{x}_{t,j}\|_2^2\right) - 2C\|\boldsymbol{\ell}^\star_t\| \quad .$$

We conclude the proof by summing the above over $1 \leq t \leq T$ and comparing the result to the upper bound in Eq. (4). □

Under the assumptions of this lemma, our algorithm competes with a set of fixed margin classifiers, $\mathbf{w}^\star_1, \ldots, \mathbf{w}^\star_k$, which may even be defined in hindsight, after observing all of the inputs and their labels. The right-hand side of the bound is the sum of two terms, a complexity term $\sum_{j=1}^{k} \|\mathbf{w}^\star_j\|_2^2$ and a term which is proportional to the cumulative loss of our competitor, $\sum_{t=1}^{T} \|\boldsymbol{\ell}^\star_t\|$. The left hand side of the bound is the term

$$\sum_{t=1}^{T}\sum_{j=1}^{k}\left(2\tau_{t,j}\ell_{t,j} - \tau^2_{t,j}\|\mathbf{x}_{t,j}\|_2^2\right) \quad . \tag{7}$$

This term plays a key role in the derivation of all three families of algorithms. As Lemma 1 provides an upper bound on Eq. (7), we prove matching lower bounds for each of our algorithms. Comparing each of these lower bounds to Lemma 1 yields a loss bound for the respective algorithm.

3 The Finite-Horizon Multitask Perceptron

Our first family of online multitask classification algorithms is called the *finite-horizon multitask Perceptron* family. This family includes algorithms for any global loss function defined by an absolute norm. These algorithms are finite-horizon online algorithms, meaning that the number of online rounds, T, is known in advance and is given as a parameter to the algorithm. An analogous family of infinite-horizon algorithms is the topic of the next section. Given an absolute norm $\|\cdot\|$ and its dual $\|\cdot\|^\star$, the multitask Perceptron sets $\tau_{t,j}$ to be

$$\boldsymbol{\tau}_t \;=\; \operatorname*{argmax}_{\boldsymbol{\tau}:\, \|\boldsymbol{\tau}\|^\star \leq C} \boldsymbol{\tau} \cdot \boldsymbol{\ell}_t \quad , \tag{8}$$

where $C > 0$ is specified later on. Using Eq. (1), we obtain the dual of $\|\cdot\|^\star$:

$$\|\boldsymbol{\ell}\|^{\star\star} \;=\; \max_{\boldsymbol{\tau}:\, \|\boldsymbol{\tau}\|^\star \leq 1} \boldsymbol{\tau} \cdot \boldsymbol{\ell} \quad .$$

Since $\|\cdot\|^{\star\star}$ and $\|\cdot\|$ are equivalent [13] and since $\|\boldsymbol{\tau}/C\|^\star = \|\boldsymbol{\tau}\|^\star/C$, we conclude that $\boldsymbol{\tau}_t$ from Eq. (8) satisfies

$$\boldsymbol{\tau}_t \cdot \boldsymbol{\ell}_t \;=\; C\|\boldsymbol{\ell}_t\| \quad . \tag{9}$$

If the global loss is a p-norm, then Eq. (8) reduces to $\tau_{t,j} = C\ell_{t,j}^{p-1}/\|\boldsymbol{\ell}_t\|_p^{p-1}$. If the global loss is an r-max norm and π is a permutation such that $\ell_{t,\pi(1)} \geq \ldots \geq \ell_{t,\pi(k)}$, then Eq. (8) reduces to

$$\tau_{t,j} \;=\; \begin{cases} C & \text{if } \; j \in \{\pi(1), \ldots, \pi(r)\} \\ 0 & \text{otherwise} \end{cases} \quad .$$

The correctness of both definitions of $\tau_{t,j}$ given above can be easily verified by observing that $\|\boldsymbol{\tau}_t\|^\star = C$ and that $\boldsymbol{\tau}_t \cdot \boldsymbol{\ell}_t = C\|\boldsymbol{\ell}_t\|$ in both cases.

An important component in our analysis is the *remoteness* of a norm $\|\cdot\|$, defined to be

$$\rho(\|\cdot\|, k) \;=\; \max_{\mathbf{u} \in \mathbb{R}^k} \frac{\|\mathbf{u}\|_2}{\|\mathbf{u}\|} \quad .$$

Geometrically, the remoteness of $\|\cdot\|$ is simply the Euclidean length of the longest vector (again, in the Euclidean sense) which is contained in the unit ball of $\|\cdot\|$. For example, for any p-norm with $p \geq 2$, $\rho(\|\cdot\|_p, k) = k^{1/2-1/p}$. In this paper, we take a specific interest in the remoteness of the dual norm $\|\cdot\|^\star$, and we abbreviate $\rho(\|\cdot\|^\star, k)$ by ρ when $\|\cdot\|^\star$ and k are obvious from the context. With this definition handy, we are ready to prove a loss bound for the multitask Perceptron.

Theorem 1. *Let $\{(\mathbf{x}_{t,j}, y_{t,j})\}_{1 \leq t \leq T}^{1 \leq j \leq k}$ be a sequence of T k-tuples of examples, where each $\mathbf{x}_{t,j} \in \mathbb{R}^{n_j}$, $\|\mathbf{x}_{t,j}\|_2 \leq R$ and each $y_{t,j} \in \{-1, +1\}$. Let $\|\cdot\|$ be an absolute norm and let ρ denote the remoteness of its dual. Let $\mathbf{w}_1^\star, \ldots, \mathbf{w}_k^\star$ be arbitrary vectors where $\mathbf{w}_j^\star \in \mathbb{R}^{n_j}$, and define the hinge loss attained by $\mathbf{w}_j^\star$ on example $(\mathbf{x}_{t,j}, y_{t,j})$ to be $\ell_{t,j}^\star = \left[1 - y_{t,j}\mathbf{w}_j^\star \cdot \mathbf{x}_{t,j}\right]_+$. If we present this sequence to the finite-horizon multitask Perceptron with the norm $\|\cdot\|$ and the parameter C, then*

$$\sum_{t=1}^{T} \|\boldsymbol{\ell}_t\| \;\leq\; \frac{1}{2C} \sum_{j=1}^{k} \|\mathbf{w}_j^\star\|_2^2 \;+\; \sum_{t=1}^{T} \|\boldsymbol{\ell}_t^\star\| \;+\; \frac{TR^2C\,\rho^2(\|\cdot\|^\star, k)}{2} \quad .$$

Proof. The starting point of our analysis is Lemma 1. The choice of $\tau_{t,j}$ in Eq. (8) is clearly bounded by $\|\boldsymbol{\tau}_t\|^\star \leq C$ and conservative. It is also non-negative, due to the fact that $\|\cdot\|^\star$ is an absolute norm and that $\ell_{t,j} \geq 0$. Therefore, the definition of $\tau_{t,j}$ in Eq. (8) meets the requirements of the lemma, and we have

$$\sum_{t=1}^{T} \sum_{j=1}^{k} \Big(2\tau_{t,j}\ell_{t,j} - \tau_{t,j}^2 \|\mathbf{x}_{t,j}\|_2^2\Big) \;\leq\; \sum_{j=1}^{k} \|\mathbf{w}_j^\star\|_2^2 \;+\; 2C \sum_{t=1}^{T} \|\boldsymbol{\ell}_t^\star\| \quad .$$

Using Eq. (9), we rewrite the left-hand side of the above as

$$2C \sum_{t=1}^{T} \|\boldsymbol{\ell}_t\| - \sum_{t=1}^{T} \sum_{j=1}^{k} \tau_{t,j}^2 \|\mathbf{x}_{t,j}\|_2^2 \quad . \tag{10}$$

Using our assumption that $\|\mathbf{x}_{t,j}\|_2^2 \leq R^2$, we know that $\sum_{j=1}^k \tau_{t,j}^2 \|\mathbf{x}_{t,j}\|_2^2 \leq R^2 \|\boldsymbol{\tau}_t\|_2^2$. Using the definition of remoteness, we can upper bound this term by $R^2(\|\boldsymbol{\tau}_t\|^\star)^2\rho^2$. Finally, using our upper bound on $\|\boldsymbol{\tau}_t\|^\star$ we can further bound this term by $R^2C^2\rho^2$. Plugging this bound back into Eq. (10) gives

$$2C\sum_{t=1}^T \|\boldsymbol{\ell}_t\| \;-\; TR^2C^2\rho^2 \quad .$$

Overall, we have shown that

$$2C\sum_{t=1}^T \|\boldsymbol{\ell}_t\| \;-\; TR^2C^2\rho^2 \;\leq\; \sum_{j=1}^k \|\mathbf{w}_j^\star\|_2^2 \;+\; 2C\sum_{t=1}^T \|\boldsymbol{\ell}_t^\star\| \quad .$$

Dividing both sides of the above by $2C$ and rearranging terms gives the desired bound. □

Corollary 1. *Under the assumptions of Thm. 1, if $C = 1/(\sqrt{T}R\rho)$, then*

$$\sum_{t=1}^T \|\boldsymbol{\ell}_t\| \;\leq\; \sum_{t=1}^T \|\boldsymbol{\ell}_t^\star\| \;+\; \frac{\sqrt{T}R\,\rho}{2}\left(\sum_{j=1}^k \|\mathbf{w}_j^\star\|_2^2 \;+\; 1\right) \quad .$$

Since our algorithm uses C in its update procedure, and C is a function of $\sqrt{T}$, then this algorithm is a finite horizon algorithm.

4 An Extension to the Infinite Horizon Setting

We would like to devise an algorithm which does not require prior knowledge of the sequence length T. Moreover, we would like a bound which holds simultaneously for every prefix of the input sequence. In this section, we adapt the multitask Perceptron to the infinite horizon setting. This generalization comes at a price; our analysis only bounds a function similar to the cumulative global loss, but not the global loss per se (see Corollary 2 below).

To motivate the infinite-horizon multitask Perceptron, we take a closer look at the analysis of the finite-horizon Perceptron, from the previous section. In the proof of Thm. 1, we lower-bounded the term $\sum_{j=1}^k 2\tau_{t,j}\ell_{t,j} - \tau_{t,j}^2\|\mathbf{x}_{t,j}\|_2^2$ by $2C\|\boldsymbol{\ell}_t\| - R^2C^2\rho^2$. The first term in this lower bound is proportional to the global loss, and the second term is a constant. When $\|\ell_t\|$ is small, the difference between these two terms may be negative, which implies that our update step-size may have been too large on that round, and that our update may have even increased our distance to the target. Here, we derive an update for which $\sum_{j=1}^k 2\tau_{t,j}\ell_{t,j} - \tau_{t,j}^2\|\mathbf{x}_{t,j}\|_2^2$ is always positive. The vector $\boldsymbol{\tau}_t$ remains in the same direction as before, but by limiting its dual norm we enforce an update step-size which is never excessively large. We replace the definition of τ_t in Eq. (8) by

$$\boldsymbol{\tau}_t \;=\; \operatorname*{argmax}_{\boldsymbol{\tau}:\, \|\boldsymbol{\tau}\|^\star \leq \min\left\{C,\, \frac{\|\boldsymbol{\ell}_t\|}{R^2\rho^2}\right\}} \boldsymbol{\tau}\cdot\boldsymbol{\ell}_t \quad , \tag{11}$$

where $C > 0$ is a user defined parameter, $R > 0$ is an upper bound on $\|\mathbf{x}_{t,j}\|_2$ for all $1 \leq t \leq T$ and $1 \leq j \leq k$, and $\rho = \rho(\|\cdot\|^\star, k)$. If the global loss function is a p-norm, then the above reduces to

$$\tau_{t,j} \;=\; \begin{cases} \frac{\ell_{t,j}^{p-1}}{R^2\rho^2\|\boldsymbol{\ell}_t\|_p^{p-2}} & \text{if } \|\boldsymbol{\ell}_t\|_p \leq R^2C\rho^2 \\ \frac{C\ell_{t,j}^{p-1}}{\|\boldsymbol{\ell}_t\|_p^{p-1}} & \text{otherwise} \end{cases} \quad .$$

If the global loss function is an r-max norm and π is a permutation such that $\ell_{t,\pi(1)} \geq \ldots \geq \ell_{t,\pi(k)}$, then Eq. (11) becomes

$$\tau_{t,j} \;=\; \begin{cases} \frac{\|\boldsymbol{\ell}_t\|_{r\text{-max}}}{rR^2} & \text{if } \|\boldsymbol{\ell}_t\|_{r\text{-max}} \leq R^2C\rho^2 \;\text{ and }\; j \in \{\pi(1), \ldots, \pi(r)\} \\ C & \text{if } \|\boldsymbol{\ell}_t\|_{r\text{-max}} > R^2C\rho^2 \;\text{ and }\; j \in \{\pi(1), \ldots, \pi(r)\} \\ 0 & \text{otherwise} \end{cases} \quad .$$

We now turn to proving a cumulative loss bound.

Theorem 2. *Let $\{(\mathbf{x}_{t,j}, y_{t,j})\}_{t=1,2,\ldots}^{1\leq j\leq k}$ be a sequence of k-tuples of examples, where each $\mathbf{x}_{t,j} \in \mathbb{R}^{n_j}$, $\|\mathbf{x}_{t,j}\|_2 \leq R$ and each $y_{t,j} \in \{-1, +1\}$. Let $\|\cdot\|$ be an absolute norm and let ρ denote the remoteness of its dual. Let $\mathbf{w}_1^\star, \ldots, \mathbf{w}_k^\star$ be arbitrary vectors where $\mathbf{w}_j^\star \in \mathbb{R}^{n_j}$, and define the hinge loss attained by $\mathbf{w}_j^\star$ on example $(\mathbf{x}_{t,j}, y_{t,j})$ to be $\ell_{t,j}^\star = \left[1 - y_{t,j}\mathbf{w}_j^\star \cdot \mathbf{x}_{t,j}\right]_+$. If we present this sequence to the infinite-horizon multitask Perceptron with the norm $\|\cdot\|$ and the parameter C, then, for every T,*

$$1/(R^2\rho^2) \sum_{t\leq T:\|\boldsymbol{\ell}_t\|\leq R^2C\rho^2} \|\boldsymbol{\ell}_t\|^2 \;+\; C \sum_{t\leq T:\|\boldsymbol{\ell}_t\|>R^2C\rho^2} \|\boldsymbol{\ell}_t\| \;\leq\; 2C\sum_{t=1}^{T} \|\boldsymbol{\ell}_t^\star\| \;+\; \sum_{j=1}^{k} \|\mathbf{w}_j^\star\|_2^2 \quad .$$

Proof. The starting point of our analysis is again Lemma 1. The choice of $\tau_{t,j}$ in Eq. (11) is clearly bounded by $\|\boldsymbol{\tau}_t\|^\star \leq C$ and conservative. It is also non-negative, due to the fact that $\|\cdot\|^\star$ is absolute and that $\ell_{t,j} \geq 0$. Therefore, $\tau_{t,j}$ meets the requirements of Lemma 1, and we have

$$\sum_{t=1}^{T}\sum_{j=1}^{k}\Big(2\tau_{t,j}\ell_{t,j} - \tau_{t,j}^2\|\mathbf{x}_{t,j}\|_2^2\Big) \;\leq\; \sum_{j=1}^{k}\|\mathbf{w}_j^\star\|_2^2 \;+\; 2C\sum_{t=1}^{T}\|\boldsymbol{\ell}_t^\star\| \quad . \tag{12}$$

We now prove our theorem by lower-bounding the left hand side of the above. We analyze two different cases: if $\|\boldsymbol{\ell}_t\| \leq R^2C\rho^2$ then $\min\{C, \|\boldsymbol{\ell}_t\|/(R^2\rho^2)\} = \|\boldsymbol{\ell}_t\|/(R^2\rho^2)$. Again using the fact that the dual of the dual norm is the original norm, together with the definition of $\boldsymbol{\tau}_t$ in Eq. (11), we get that

$$2\sum_{j=1}^{k}\tau_{t,j}\ell_{t,j} \;=\; 2\|\boldsymbol{\tau}_t\|^\star\,\|\boldsymbol{\ell}_t\| \;=\; 2\frac{\|\boldsymbol{\ell}_t\|^2}{R^2\rho^2} \quad . \tag{13}$$

On the other hand, $\sum_{j=1}^{k} \tau_{t,j}^2 \|\mathbf{x}_{t,j}\|_2^2$ can be bounded by $R^2 \|\boldsymbol{\tau}_t\|_2^2$. Using the definition of remoteness, we bound this term by $R^2(\|\boldsymbol{\tau}_t\|^\star)^2\rho^2$. Using the fact that, $\|\boldsymbol{\tau}_t\|^\star \leq \|\boldsymbol{\ell}_t\|/(R^2\rho^2)$, we bound this term by $\|\boldsymbol{\ell}_t\|^2/(R^2\rho^2)$. Overall, we have shown that $\sum_{j=1}^{k} \tau_{t,j}^2 \|\mathbf{x}_{t,j}\|_2^2 \leq \frac{\|\boldsymbol{\ell}_t\|^2}{R^2\rho^2}$. Subtracting both sides of this inequality from the respective sides of Eq. (13) gives

$$\frac{\|\boldsymbol{\ell}_t\|^2}{R^2\rho^2} \;\leq\; \sum_{j=1}^{k} \left(2\tau_{t,j}\ell_{t,j} - \tau_{t,j}^2 \|\mathbf{x}_{t,j}\|_2^2\right) \quad . \tag{14}$$

Moving on to the second case, if $\|\boldsymbol{\ell}_t\| > R^2 C \rho^2$ then $\min\{C, \|\boldsymbol{\ell}_t\|/(R^2\rho^2)\} = C$. As in Eq. (9), we have that

$$2\sum_{j=1}^{k} \tau_{t,j}\ell_{t,j} \;=\; 2\|\boldsymbol{\tau}_t\|^\star \|\boldsymbol{\ell}_t\| \;=\; 2C\|\boldsymbol{\ell}_t\| \quad . \tag{15}$$

As before, we can upper bound $\sum_{j=1}^{k} \tau_{t,j}^2 \|\mathbf{x}_{t,j}\|_2^2$ by $R^2(\|\boldsymbol{\tau}_t\|^\star)^2\rho^2$. Using the fact that $\|\boldsymbol{\tau}_t\|^\star \leq C$, we bound this term by $R^2C^2\rho^2$. Finally, using our assumption that $\|\boldsymbol{\ell}_t\| > R^2C\rho^2$, we conclude that $\sum_{j=1}^{k} \tau_{t,j}^2 \|\mathbf{x}_{t,j}\|_2^2 < C\|\boldsymbol{\ell}_t\|$. Subtracting both sides of this inequality from the respective sides of Eq. (15) gives

$$C\|\boldsymbol{\ell}_t\| \;\leq\; \sum_{j=1}^{k} \left(2\tau_{t,j}\ell_{t,j} - \tau_{t,j}^2 \|\mathbf{x}_{t,j}\|_2^2\right) \quad . \tag{16}$$

Comparing the upper bound in Eq. (12) with the lower bounds in Eq. (14) and Eq. (16) proves the theorem. □

Corollary 2. *Under the assumptions of Thm. 2, if C is set to be $1/(R^2\rho^2)$ then, for every T, it holds that*

$$\sum_{t=1}^{T} \min\left\{\|\boldsymbol{\ell}_t\|^2, \|\boldsymbol{\ell}_t\|\right\} \;\leq\; 2\sum_{t=1}^{T} \|\boldsymbol{\ell}_t^\star\| \;+\; R^2\rho^2 \sum_{j=1}^{k} \|\mathbf{w}_j^\star\|_2^2 \quad .$$

5 The Implicit Online Multitask Update

We now discuss a third family of online multitask algorithms, which leads to the strongest loss bounds of the three families of algorithms presented in this paper. In contrast to the closed form updates of the previous algorithms, the algorithms in this family require solving an optimization problem on every round, and are therefore called *implicit* update algorithms. This optimization problem captures the fundamental tradeoff inherent to online learning. On one hand, the algorithm wants its next set of hypotheses to remain close to the current set of hypotheses, so as to maintain the information learned so far. On the other

input: aggressiveness parameter $C > 0$, norm $\|\cdot\|$

initialize $\mathbf{w}_{1,1} = \ldots = \mathbf{w}_{1,k} \;=\; (0, \ldots, 0)$

for $t = 1, 2, \ldots$

- receive $\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,k}$
- predict $\;\text{sign}(\mathbf{w}_{t,j} \cdot \mathbf{x}_{t,j})$ $\qquad [1 \le j \le k]$
- receive $y_{t,1}, \ldots, y_{t,k}$
- suffer loss $\ell_{t,j} = \left[1 - y_{t,j}\mathbf{w}_{t,j} \cdot \mathbf{x}_{t,j}\right]_+$ $\qquad [1 \le j \le k]$
- update:

$$\{\mathbf{w}_{t+1,1}, \ldots, \mathbf{w}_{t+1,k}\} \;=\; \operatorname*{argmin}_{\mathbf{w}_1,\ldots,\mathbf{w}_k,\boldsymbol{\xi}} \; \tfrac{1}{2}\sum_{j=1}^{k} \|\mathbf{w}_j - \mathbf{w}_{t,j}\|_2^2 + C\|\boldsymbol{\xi}\|$$
$$\text{s.t.} \;\; \forall j \quad \mathbf{w}_j \cdot \mathbf{x}_{t,j} \ge 1 - \xi_j \;\; \text{and} \;\; \xi_j \ge 0$$

Fig. 2. The implicit update algorithm

hand, the algorithm wants to make progress using the new examples obtained on this round, where progress is measured using the global loss function. The pseudo-code of the implicit update algorithm is presented in Fig. 2.

Next, we find the dual of the optimization problem given in Fig. 2. By doing so, we show that the family of implicit update algorithms follows the skeleton outlined in Fig. 1 and satisfies the requirements of Lemma 1.

Lemma 2. *Let $\|\cdot\|$ be a norm and let $\|\cdot\|^\star$ be its dual. Then the online update defined in Fig. 2 is conservative and equivalent to setting $\mathbf{w}_{t+1,j} = \mathbf{w}_{t,j} + \tau_{t,j} y_{t,j} \mathbf{x}_{t,j}$ for all $1 \le j \le k$, where*

$$\boldsymbol{\tau}_t \;=\; \operatorname*{argmax}_{\boldsymbol{\tau}} \; \sum_{j=1}^{k} \left(2\tau_j \ell_{t,j} - \tau_j^2 \|\mathbf{x}_{t,j}\|_2^2\right) \quad \text{s.t.} \quad \|\boldsymbol{\tau}\|^\star \le C \;\; \text{and} \;\; \forall j \;\; \tau_j \ge 0 \;\; .$$

Proof Sketch. The update step in Fig. 2 sets the vectors $\mathbf{w}_{t+1,1}, \ldots, \mathbf{w}_{t+1,k}$ to be the solution to the following constrained minimization problem:

$$\min_{\mathbf{w}_1,\ldots,\mathbf{w}_k,\boldsymbol{\xi}\ge 0} \; \frac{1}{2}\sum_{j=1}^{k} \|\mathbf{w}_j - \mathbf{w}_{t,j}\|_2^2 \;+\; C\|\boldsymbol{\xi}\| \quad \text{s.t.} \quad \forall j \;\; y_{t,j}\mathbf{w}_j \cdot \mathbf{x}_{t,j} \;\ge\; 1 - \xi_j \;\; .$$

We use the notion of strong duality to restate this optimization problem in an equivalent form. The objective function above is convex and the constraints are both linear and feasible, therefore Slater's condition [14] holds, and the above problem is equivalent to

$$\max_{\boldsymbol{\tau}\ge 0} \min_{\mathbf{w}_1,\ldots,\mathbf{w}_k,\boldsymbol{\xi}\ge 0} \; \frac{1}{2}\sum_{j=1}^{k} \|\mathbf{w}_j - \mathbf{w}_{t,j}\|_2^2 + C\|\boldsymbol{\xi}\| + \sum_{j=1}^{k} \tau_j \left(1 - y_{t,j}\mathbf{w}_j \cdot \mathbf{x}_{t,j} - \xi_j\right) \;\; .$$

We can write the objective function above as the sum of two separate terms,

$$\underbrace{\frac{1}{2}\sum_{j=1}^{k}\|\mathbf{w}_j-\mathbf{w}_{t,j}\|_2^2+\sum_{j=1}^{k}\tau_j(1-y_{t,j}\mathbf{w}_j\cdot\mathbf{x}_{t,j})}_{\mathcal{L}_1(\boldsymbol{\tau},\mathbf{w}_1,\ldots,\mathbf{w}_k)} \;+\; \underbrace{C\|\boldsymbol{\xi}\|-\sum_{j=1}^{k}\tau_j\xi_j}_{\mathcal{L}_2(\boldsymbol{\tau},\boldsymbol{\xi})} \quad .$$

Using the notation defined above, our optimization problem becomes,

$$\max_{\boldsymbol{\tau}\geq 0}\left(\min_{\mathbf{w}_1,\ldots,\mathbf{w}_k}\mathcal{L}_1(\boldsymbol{\tau},\mathbf{w}_1,\ldots,\mathbf{w}_k)\;+\;\min_{\boldsymbol{\xi}\geq 0}\mathcal{L}_2(\boldsymbol{\tau},\boldsymbol{\xi})\right) \quad .$$

For any choice of $\boldsymbol{\tau}$, $\mathcal{L}_1$ is a convex function and we can find $\mathbf{w}_1,\ldots,\mathbf{w}_k$ which minimize it by setting all of its partial derivatives with respect to the elements of $\mathbf{w}_1,\ldots,\mathbf{w}_k$ to zero. By doing so, we conclude that, $\mathbf{w}_j=\mathbf{w}_{t,j}+\tau_j y_{t,j}\mathbf{x}_{t,j}$ for all $1\leq j\leq k$.

The update is conservative since if $\ell_{t,j}=0$ then setting $\mathbf{w}_j=\mathbf{w}_{t,j}$ satisfies the constraints and minimizes $\|\mathbf{w}_t-\mathbf{w}_{t,j}\|_2^2$, without restricting our choice of any other variable. Plugging our expression for $\mathbf{w}_j$ into $\mathcal{L}_1$, we have that

$$\min_{\mathbf{w}_1,\ldots,\mathbf{w}_k}\mathcal{L}_1(\boldsymbol{\tau},\mathbf{w}_1,\ldots,\mathbf{w}_k)\;=\;\sum_{j=1}^{k}\tau_j(1-y_{t,j}\mathbf{w}_{t,j}\cdot\mathbf{x}_{t,j})\;-\;\frac{1}{2}\sum_{j=1}^{k}\tau_j^2\|\mathbf{x}_{t,j}\| \quad .$$

Since the update is conservative, it holds that $\tau_j(1-y_{t,j}\mathbf{w}_{t,j}\cdot\mathbf{x}_{t,j})=\tau_t\ell_{t,j}$. Overall, we have reduced our optimization problem to

$$\boldsymbol{\tau}_t\;=\;\operatorname*{argmax}_{\boldsymbol{\tau}\geq 0}\left(\sum_{j=1}^{k}\Big(\tau_j\ell_{t,j}-\frac{1}{2}\tau_j^2\|\mathbf{x}_{t,j}\|\Big)\;+\;\min_{\boldsymbol{\xi}\geq 0}\mathcal{L}_2(\boldsymbol{\tau},\boldsymbol{\xi})\right) \quad . \tag{17}$$

We turn our attention to $\mathcal{L}_2$ and abbreviate $B(\boldsymbol{\tau})=\min_{\boldsymbol{\xi}\geq 0}\mathcal{L}_2(\boldsymbol{\tau},\boldsymbol{\xi})$. It can now be shown that B is a barrier function for the constraint $\|\boldsymbol{\tau}\|^\star\leq C$, namely $B(\boldsymbol{\tau})=0$ if $\|\boldsymbol{\tau}\|^\star\leq C$ and $B(\boldsymbol{\tau})=-\infty$ if $\|\boldsymbol{\tau}\|^\star>C$. Therefore, we replace $B(\boldsymbol{\tau})$ in Eq. (17) with the explicit constraint $\|\boldsymbol{\tau}\|^\star\leq C$, and conclude the proof. □

Turning to the analysis, we now show that all of the loss bounds proven in this paper also apply to the implicit update family of algorithms. We formally prove that the bound in Thm. 1 (and specifically in Corollary 1) holds for this family. The proof that the bound in Thm. 2 (and specifically in Corollary 2) also holds for this family is identical and is therefore omitted.

Theorem 3. *The bound in Thm. 1 also holds for the algorithm in Fig. 2.*

Proof. Let $\tau'_{t,j}$ denote the weights defined by the multitask Perceptron in Eq. (8) and let $\tau_{t,j}$ denote the weights assigned by the implicit update in Fig. 2. In the proof of Thm. 1, we showed that,

$$2C\|\boldsymbol{\ell}_t\|-R^2C^2\rho^2\;\leq\;\sum_{j=1}^{k}\left(2\tau'_{t,j}\ell_{t,j}-\tau'^{\,2}_{t,j}\|\mathbf{x}_{t,j}\|_2^2\right) \quad . \tag{18}$$

According to Lemma 2, the weights $\tau_{t,j}$ maximize $\sum_{j=1}^{k}(2\tau_{t,j}\ell_{t,j} - \tau_{t,j}^2\|\mathbf{x}_{t,j}\|_2^2)$, subject to the constraints $\|\boldsymbol{\tau}_t\|^\star \leq C$ and $\tau_{t,j} \geq 0$. Since the weights $\tau'_{t,j}$ also satisfy these constraints, it holds that $\sum_{j=1}^{k}(2\tau'_{t,j}\ell_{t,j} - \tau'^2_{t,j}\|\mathbf{x}_{t,j}\|_2^2)$ is necessarily upper-bounded by $\sum_{j=1}^{k}(2\tau_{t,j}\ell_{t,j} - \tau_{t,j}^2\|\mathbf{x}_{t,j}\|_2^2)$. Combining this fact with Eq. (18), we conclude that

$$2C\|\boldsymbol{\ell}_t\| - R^2C^2\rho^2 \;\leq\; \sum_{j=1}^{k}\left(2\tau_{t,j}\ell_{t,j} - \tau_{t,j}^2\|\mathbf{x}_{t,j}\|_2^2\right) \quad . \tag{19}$$

Since $\tau_{t,j}$ is bounded, non-negative, and conservative (due to Lemma 2), the right-hand side of the above inequality is upper-bounded by Lemma 1. Comparing the bound in Eq. (19) to the bound in Lemma 1 proves the theorem. □

6 Algorithms for the Implicit Online Multitask Update

In this section, we briefly describe efficient algorithms for calculating the update in Fig. 2. Due to space constraints we omit all proofs. If the global loss function is the L_2 norm, it can be shown that the solution to the optimization problem in Fig. 2 takes the form $\tau_{t,j} = \ell_{t,j}/(\|\mathbf{x}_{t,j}\|_2^2 + \theta_t)$, where θ_t is the solution to the equation $\sum_{j=1}^{k}(\frac{\ell_{t,j}}{\|\mathbf{x}_{t,j}\|_2^2+\theta_t})^2 = C^2$. The exact value of θ_t can be found using binary search.

Similarly, in the case where the global loss function is the r-max norm, it can be shown that there exists some $\theta_t \geq 0$ such that

$$\tau_{t,j} \;=\; \begin{cases} 0 & \text{if } \ell_{t,j} - \theta_t < 0 \\ \frac{\ell_{t,j}-\theta_t}{\|\mathbf{x}_{t,j}\|_2^2} & \text{if } 0 \leq \ell_{t,j} - \theta_t \leq C\|\mathbf{x}_{t,j}\|_2^2 \\ C & \text{if } C\|\mathbf{x}_{t,j}\|_2^2 < \ell_{t,j} - \theta_t \end{cases} \quad . \tag{20}$$

That is, if the loss of task j is very small then the j'th predictor is left intact. If this loss is moderate then the size of the update step for the j'th predictor is proportional to the loss suffered by the j'th task, and inversely proportional to the squared norm of $\mathbf{x}_{t,j}$. In any case, the size of the update step does not exceed the fixed upper limit C. By plugging Eq. (20) back into the objective function of our optimization problem, we can see that the objective function is monotonically decreasing in θ_t. We conclude that θ_t should be the smallest non-negative value for which the resulting update vector $\boldsymbol{\tau}_t$ satisfies the constraint $\|\boldsymbol{\tau}_t\|^\star_{r\text{-max}} \leq C$.

First, we check whether the constraint $\|\boldsymbol{\tau}_t\|^\star_{r\text{-max}} \leq C$ holds when $\theta_t = 0$. If the answer is positive, we are done. If the answer is negative, the definition of $\|\cdot\|^\star_{r\text{-max}}$ in Eq. (2) and the KKT conditions of optimality yield that $\|\boldsymbol{\tau}_t\|_1 = rC$. This equality enables us to narrow the search for θ_t to a handful of candidate values. To see this, assume for a moment that we have some way of obtaining the sets $\Psi = \{1 \leq j \leq k \,:\, 0 < \ell_{t,j} - \theta_t\}$ and $\Phi = \{1 \leq j \leq k \,:\, C\|\mathbf{x}_{t,j}\|_2^2 < \ell_{t,j} - \theta_t\}$. The semantics of Ψ and Φ are readily available from Eq. (20): the set Ψ consists

of all indices j for which $\tau_{t,j} > 0$, while Φ consists of all indices j for which $\tau_{t,j}$ is capped at C. Given these two sets and the fact that $\sum_{j=1}^{k} \tau_j = rC$ yields that

$$\sum_{j \in \Psi \setminus \Phi} \frac{\ell_{t,j} - \theta_t}{\|\mathbf{x}_{t,j}\|_2^2} \; + \; \sum_{j \in \Phi} C \;\; = \;\; rC \;\; .$$

Solving the above equation for θ_t gives

$$\theta_t \;\; = \;\; \frac{\sum_{j \in \Psi \setminus \Phi} \frac{\ell_{t,j}}{\|\mathbf{x}_{t,j}\|_2^2} - rC + \sum_{j \in \Phi} C}{\sum_{j \in \Psi \setminus \Phi} \frac{1}{\|\mathbf{x}_{t,j}\|_2^2}} \;\; . \tag{21}$$

Therefore, our problem has reduced to the problem of finding the sets Ψ and Φ.

Let $q_1, \ldots, q_{2k}$ denote the sequence of numbers obtained by sorting the union of the sets $\{\ell_{t,j}\}_{j=1}^{k}$ and $\{(\ell_{t,j} - C\|\mathbf{x}_{t,j}\|_2^2)\}_{j=1}^{k}$ in ascending order. For every $1 \leq s \leq 2k$, we define the sets $\Psi_s = \{1 \leq j \leq k : 0 < \ell_{t,j} - q_s\}$ and $\Phi_s = \{1 \leq j \leq k : C\|\mathbf{x}_{t,j}\|_2^2 < \ell_{t,j} - q_s\}$. It is not difficult to see that if $\theta_t \in [q_s, q_{s+1})$ then $\Psi = \Psi_s$ and $\Phi = \Phi_s$. Essentially, we have narrowed the search for θ_t to $2k$ candidates defined by the sets Ψ_s and Φ_s for $1 \leq s \leq 2k$, and by Eq. (21). Of these candidates we choose the smallest one which results in an update that satisfies our constraints. When performing this process with careful bookkeeping, calculating the update takes only $O(k \log(k))$ operations.

Acknowledgement This research was funded by the Israeli Science Foundation under grant number 522/04 and by the National Science Foundation ITR program under grant number 0205594.

References

1. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive aggressive algorithms. Journal of Machine Learning Research **7** (2006)
2. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. JAIR **2** (1995) 263–286
3. Caruana, R.: Multitask learning. Machine Learning **28** (1997) 41–75
4. Heskes, T.: Solving a huge number of silmilar tasks: A combination of multitask learning and a hierarchical bayesian approach. In: ICML 15. (1998) 233–241
5. Evgeniou, T., C.Micchelli, Pontil, M.: Learning multiple tasks with kernel methods. Journal of Machine Learning Research **6** (2005) 615–637
6. Baxter, J.: A model of inductive bias learning. Journal of Artificial Intelligence Research **12** (2000) 149–198
7. Ben-David, S., Schuller, R.: Exploiting task relatedness for multiple task learning. In: COLT 16. (2003)
8. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: Proceedings of the Twenty-First International Conference on Machine Learning. (2004)
9. Kivinen, J., Warmuth, M.: Relative loss bounds for multidimensional regression problems. Journal of Machine Learning **45** (2001) 301–329

10. Helmbold, D., Kivinen, J., Warmuth, M.: Relative loss bounds for single neurons. IEEE Transactions on Neural Networks **10** (1999) 1291–1304
11. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. Jornal of Machine Learning Research **2** (2001) 265–292
12. Crammer, K., Singer, Y.: Ultraconservative online algorithms for multiclass problems. Jornal of Machine Learning Research **3** (2003) 951–991
13. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge Univ. Press (1985)
14. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge Univ. Press (2004)