Journal of Machine Learning Research 8 (2007) 2233-2264 Submitted 9/06; Revised 9/07; Published 10/07

Online Learning of Multiple Tasks with a Shared Loss

Ofer Dekel OFERD@CS.HUJI.AC.IL
School of Computer Science and Engineering
The Hebrew University
Jerusalem, 91904, Israel

Philip M. Long PLONG@GOOGLE.COM
Yoram Singer SINGER@GOOGLE.COM
Google Inc.
1600 Amphitheater Parkway
Mountain View, CA 94043, USA

Editor: Peter Bartlett

Abstract

We study the problem of learning multiple tasks in parallel within the online learning framework. On each online round, the algorithm receives an instance for each of the parallel tasks and responds by predicting the label of each instance. We consider the case where the predictions made on each round all contribute toward a common goal. The relationship between the various tasks is defined by a global loss function, which evaluates the overall quality of the multiple predictions made on each round. Specifically, each individual prediction is associated with its own loss value, and then these multiple loss values are combined into a single number using the global loss function. We focus on the case where the global loss function belongs to the family of absolute norms, and present several online learning algorithms for the induced problem. We prove worst-case relative loss bounds for all of our algorithms, and demonstrate the effectiveness of our approach on a large-scale multiclass-multilabel text categorization problem.

Keywords: online learning, multitask learning, multiclass multilabel classiifcation, perceptron

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, and focus on the possibility that the learning tasks contribute toward a common goal. Our hope is that we can benefit from learning 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 and multiclass problems using ideas in Crammer et al. (2006). In the online multitask classification setting, we are faced with k separate online binary classification problems, which are presented to us 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 k instances, and then receives the k correct labels. At this point, each of the algorithm's predictions is associated with a non-negative loss, and we use $\ell_t = (\ell_{t,1}, \ldots, \ell_{t,k})$ to denote the k-coordinate vector whose elements are the individual loss values associated with the respective tasks. Let $\mathcal{L} : \mathbb{R}^k \to \mathbb{R}_+$ be a predetermined *global loss* function, which is used to combine the individual loss values into a single number, and define the global loss attained on round t to be $\mathcal{L}(\ell_t)$. At the end of this 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, $\sum_{t=1}^{T} \mathcal{L}(\ell_t)$.

The choice of the global loss function captures the overall consequences of the individual prediction errors, and therefore how the algorithm should prioritize correcting errors. For example, if $\mathcal{L}(\ell_t)$ is defined to be $\sum_{j=1}^{k} \ell_{t,j}$ then the online algorithm is penalized equally for errors on each of the tasks; this results in effectively treating the tasks independently. On the other hand, if $\mathcal{L}(\ell_t) = \max_j \ell_{t,j}$ then the algorithm is only interested in the worst mistake made on each round. We do not assume that the data sets of the various tasks are similar or otherwise related. Moreover, the examples presented to the algorithm for each of the tasks may come from completely 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

$$\|\mathbf{v}\|_p = \Big(\sum_{j=1}^{n} |v_j|^p\Big)^{1/p} .$$

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 known 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, the r-max norm is

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

Note that both the L_1 norm and L_∞ norm are special cases of the r-max norm, as well as being p-norms. Actually, the r-max norm can be viewed as a smooth interpolation between the L_1 norm and the L_∞ norm, using Peetre's K-method of norm interpolation (see Appendix A for details).

Since the global loss functions we consider in this paper are norms, the global loss equals zero only if ℓ_t is itself the zero vector. Furthermore, decreasing any individual loss can only decrease the global loss function. Therefore, the simplest solution to our multitask problem is to learn each task individually, and minimize the global loss function implicitly. The natural question which is at the heart of this paper is whether we can do better than this. Our answer to this question is based on the following fundamental view of online learning. On every round, the online learning algorithm balances a trade-off between retaining the information it has acquired on previous rounds and modifying its hypothesis based on the new examples obtained on that round. Instead of balancing this trade-off individually for each of the learning tasks, we can balance it jointly, for all of the tasks. By doing so, we allow ourselves to make a big modification to one of the k hypotheses at the expense of the others. This additional flexibility enables us to directly minimize the specific global loss function we have chosen to use.

To motivate and demonstrate the practicality of our approach, we begin with 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. This approach is often called the *one-vs-rest* method. If all of the binary classifiers make correct predictions, then one of these predictions should be positive and the rest should be negative. If this is the case, we can correctly predict the corresponding multiclass label. However, if one or more of the binary classifiers makes an incorrect prediction, we can no longer guarantee the correctness of our multiclass prediction. In this sense, a single binary mistake on round t is as bad as many binary mistakes on round t. Therefore, we should only care about the worst binary prediction on round t, and we can do so by choosing the global loss to be $\|\ell_t\|_\infty$.

Another example where the L_∞ norm comes in handy is the case where we are faced with a multiclass problem where the number of labels is huge. Specifically, we would like the running time and the space complexity of our algorithm to scale logarithmically with the number of labels. Assume that the number of different labels is 2^k, enumerate these labels from 0 to 2^k-1, and consider the k-bit binary representation of each label. We can solve the multiclass problem by training k binary classifiers, one for each bit in the binary representation of the label index. If all k classifiers make correct predictions, then we have obtained the binary representation of the correct multiclass label. As before, a single binary mistake is devastating to the multiclass classifier, and the L_∞ norm is the most appropriate means of combining the k individual losses into a global loss.

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 choice of the global loss function is the L_2 norm, which reduces 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 (ECOC) is a technique for reducing a multiclass classification problem to multiple binary classification problems (Dietterich and Bakiri, 1995). 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 $M \in \{-1,+1\}^{s\times k}$, 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 in Equation (1), can serve as a good proxy.

In this paper, we present three families of online multitask algorithms. Each family includes algorithms for every absolute norm. All of the algorithms presented in this paper follow the general skeleton outlined in Figure 1. Specifically, all of our algorithms use linear threshold functions as hypotheses and an additive update rule. The first two families are multitask extensions of the Perceptron algorithm (Rosenblatt, 1958; Novikoff, 1962), while the third family is closely related to the Passive-Aggressive classification algorithm (Crammer et al., 2006). Incidentally, all of the algorithms presented in this paper can be easily transformed 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 comparable to the cumulative loss attained by any fixed set of k linear hypotheses, even defined in hindsight.

Much previous work on theoretical and applied multitask learning has focused on how to take advantage of similarities between the various tasks (Caruana, 1997; Heskes, 1998; Evgeniou et al., 2005; Baxter, 2000; Ben-David and Schuller, 2003; Tsochantaridis et al., 2004); in contrast, we do not assume that the tasks are in any way related. Instead, we consider how to take account of shared consequences of errors. Kivinen and Warmuth (2001) generalized the notion of matching loss (Helmbold et al., 1999) to multi-dimensional outputs. Their construction 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 problems that fall into our setting. An analysis of the L_∞ norm of prediction errors is implicit in some past work of Crammer and Singer (2001, 2003). The algorithms presented in Crammer and Singer (2001, 2003) were devised for multiclass categorization with multiple predictors (one per class) and a single instance. The present paper extends the multiclass prediction setting to a broader framework, and tightens the analysis. In contrast to the multiclass prediction setting, the prediction tasks in our setting are tied solely through a globally shared loss. When k, the number of multiple tasks, is set to 1, two of the algorithms presented in this paper as well as the multiclass algorithms in Crammer and Singer (2001, 2003) reduce to the PA-I algorithm, presented in Crammer et al. (2006). Last, we would like to mention in passing that a few learning algorithms for ranking problems decompose the ranking problem into a preference learning task over pairs of instances (see for instance Herbrich et al., 2000; Chapelle and Harchaoui, 2005). The ranking losses employed by such algorithms are typically defined as the sum over pair-based losses. Our setting generalizes such approaches for ranking learning by employing a shared loss which is defined through a norm over the individual pair-based losses.

This paper is organized as follows. In Section 2 we present our problem more formally and prove a key lemma which facilitates the analysis of our algorithms. In Section 3 we present our first family of algorithms, which works in the finite-horizon online setting. In Section 4 we extend the first family of algorithms to the infinite-horizon online setting. Then, in Section 5 we present our third family of algorithms, and show that it shares the analyses of both previous families. The third family of algorithms requires solving a small optimization problem on each online round, and is therefore called the *implicit update* family of algorithms. In Section 6 and Section 7 we describe

```
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 $\mathrm{sign}(\mathbf{w}_{t,j} \cdot \mathbf{x}_{t,j})$                         $[1 \le j \le k]$
    • 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 $\mathbf{w}_{t+1,j} = \mathbf{w}_{t,j} + \tau_{t,j} y_{t,j} \mathbf{x}_{t,j}$                  $[1 \le j \le k]$
```

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

efficient algorithms for solving the implicit update in the case where the global loss is defined by the L_2 norm or the r-max norm. Experimental results are provided in Section 8 and we conclude the paper in Section 9 with a short discussion.

2. Online Multitask Learning with Additive Updates

We begin by presenting the online multitask classification setting more formally. We are presented with k online binary classification problems in parallel. The instances of each task are drawn from separate instance domains, and for concreteness we assume that the instances of task 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, which are updated from round to round. Each of these classifiers is a margin-based linear predictor, defined by a weight vector. We denote the weight vector used on round t to define the j'th predictor by $\mathbf{w}_{t,j}$ and note that $\mathbf{w}_{t,j} \in \mathbb{R}^{n_j}$. The algorithm uses its classifiers to make k binary predictions, $\hat{y}_{t,1},\ldots,\hat{y}_{t,k}$, where $\hat{y}_{t,j} = \mathrm{sign}(\mathbf{w}_{t,j} \cdot \mathbf{x}_{t,j})$. After making these predictions, the correct labels of the respective tasks, $y_{t,1},\ldots,y_{t,k}$, are revealed and each one of the predictions is evaluated. In this paper we focus on the hinge-loss function as the means of penalizing incorrect predictions. Formally, the loss associated with the j'th task 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 $\|\ell_t\|$, 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 scalar. The algorithms only differ from one another in the specific way in which $\tau_{t,j}$ is set. For convenience, we denote $\tau_t = (\tau_{t,1}, \ldots, \tau_{t,k})$. The general skeleton followed by all of our online algorithms is given in Figure 1.

A concept of key importance in this paper is the notion of *dual norms* (Horn and Johnson, 1985). Any norm $\|\cdot\|$ defined on $\mathbb{R}^n$, has a dual norm, also defined on $\mathbb{R}^n$, denoted by $\|\cdot\|^*$ and given by

$$\|\mathbf{u}\|^* \;=\; \max_{\mathbf{v}\in\mathbb{R}^n} \frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{v}\|} \;=\; \max_{\mathbf{v}\in\mathbb{R}^n : \|\mathbf{v}\|=1} \mathbf{u}\cdot\mathbf{v} \;\; . \tag{2}$$

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. In Appendix A we prove that the dual of $\|\mathbf{v}\|_{r\text{-max}}$ is

$$\|\mathbf{u}\|^*_{r\text{-max}} \;=\; \max\left\{ \|\mathbf{u}\|_\infty, \frac{\|\mathbf{u}\|_1}{r} \right\} \;\; . \tag{3}$$

An important property of dual norms, which is an immediate consequence of Equation (2), is that for any $\mathbf{u},\mathbf{v}\in\mathbb{R}^n$ it holds that

$$\mathbf{u}\cdot\mathbf{v} \;\le\; \|\mathbf{u}\|^* \, \|\mathbf{v}\| \;\; . \tag{4}$$

If $\|\cdot\|$ is a p-norm then the above is known as Hölder's inequality, and specifically, if $p=2$ 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 Horn and Johnson, 1985), 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 $\tau_{t,j}$ for each task on each round. The different ways of setting $\tau_{t,j}$ discussed in this paper all share the following properties:

- *boundedness:* $\forall\, 1 \le t \le T \quad \|\tau_t\|^* \le C$ for some predefined parameter C
- *non-negativity:* $\forall\, 1 \le t \le T,\; 1 \le j \le k \quad \tau_{t,j} \ge 0$
- *conservativeness:* $\forall\, 1 \le t \le T,\; 1 \le j \le k \quad (\ell_{t,j} = 0) \;\Rightarrow\; (\tau_{t,j} = 0)$

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 $\mathbf{x}_{t,j}\in\mathbb{R}^{n_j}$, 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 an arbitrary norm and let $\|\cdot\|^*$ denote its dual. Assume we apply an algorithm of the form outlined in Figure 1 to this sequence of examples, where the update weights satisfy the boundedness, non-negativity and conservativeness requirements. Then, for any $C > 0$ it holds that*

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

Under the assumptions of this lemma, our algorithm competes with a set of fixed linear classifiers, $\mathbf{w}_1^\star,\ldots,\mathbf{w}_k^\star$, 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 $\sum_{j=1}^{k}\|\mathbf{w}_j^\star\|_2^2$ and a term which is proportional to the cumulative loss of our competitor, $\sum_{t=1}^{T}\|\ell_t^\star\|$. The left hand side of the bound is the term

$$\sum_{t=1}^{T}\sum_{j=1}^{k}\left(2\tau_{t,j}\ell_{t,j}-\tau_{t,j}^2\|\mathbf{x}_{t,j}\|_2^2\right) \ . \tag{5}$$

This term plays a key role in the derivation of all three families of algorithms presented in the sequel. Each choice of the update weights $\tau_{t,j}$ enables us to prove a different lower bound on Equation (5). Comparing this lower bound with the upper bound in Lemma 1 gives us a loss bound for the respective algorithm. The proof of Lemma 1 is given below.

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 $1\leq j\leq k$, $\sum_{t=1}^{T}\Delta_{t,j}$ is a telescopic sum which collapses to

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

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

$$\sum_{t=1}^{T}\sum_{j=1}^{k}\Delta_{t,j} \ \leq \ \sum_{j=1}^{k}\|\mathbf{w}_j^\star\|_2^2 \ . \tag{6}$$

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

$$\begin{aligned}\Delta_{t,j} &= \|\mathbf{w}_{t,j}-\mathbf{w}_j^\star\|_2^2-\|\mathbf{w}_{t,j}+\tau_{t,j}y_{t,j}\mathbf{x}_{t,j}-\mathbf{w}_j^\star\|_2^2\\ &= \tau_{t,j}\left(-2y_{t,j}\mathbf{w}_{t,j}\cdot\mathbf{x}_{t,j}-\tau_{t,j}\|\mathbf{x}_{t,j}\|_2^2+2y_{t,j}\mathbf{w}_j^\star\cdot\mathbf{x}_{t,j}\right)\\ &= \tau_{t,j}\left(2(1-y_{t,j}\mathbf{w}_{t,j}\cdot\mathbf{x}_{t,j})-\tau_{t,j}\|\mathbf{x}_{t,j}\|_2^2-2(1-y_{t,j}\mathbf{w}_j^\star\cdot\mathbf{x}_{t,j})\right) \ .\end{aligned} \tag{7}$$

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 $\ell_{t,j}^\star\geq 1-y_{t,j}\mathbf{w}_j^\star\cdot\mathbf{x}_{t,j}$. Plugging these two facts into Equation (7) and using the fact that $\tau_{t,j}$ is non-negative gives

$$\Delta_{t,j} \ \geq \ \tau_{t,j}\left(2\ell_{t,j}-\tau_{t,j}\|\mathbf{x}_{t,j}\|_2^2-2\ell_{t,j}^\star\right) \ .$$

Summing the above over $1\leq j\leq k$ gives

$$\sum_{j=1}^{k}\Delta_{t,j} \ \geq \ \sum_{j=1}^{k}\left(2\tau_{t,j}\ell_{t,j}-\tau_{t,j}^2\|\mathbf{x}_{t,j}\|_2^2\right)-2\sum_{j=1}^{k}\tau_{t,j}\ell_{t,j}^\star \ . \tag{8}$$

Using Equation (4) we know that $\sum_{j=1}^{k}\tau_{t,j}\ell_{t,j}^\star\leq\|\tau_t\|^*\|\ell_t^\star\|$. From our assumption that $\|\tau_t\|^*\leq C$, we have that $\sum_{j=1}^{k}\tau_{t,j}\ell_{t,j}^\star\leq C\|\ell_t^\star\|$. Plugging this inequality into Equation (8) gives

$$\sum_{j=1}^{k}\Delta_{t,j} \ \geq \ \sum_{j=1}^{k}\left(2\tau_{t,j}\ell_{t,j}-\tau_{t,j}^2\|\mathbf{x}_{t,j}\|_2^2\right)-2C\|\ell_t^\star\| \ .$$

We conclude the proof by summing the above over $1\leq t\leq T$ and comparing the result to the upper bound in Equation (6). ■

3. The Finite-Horizon Multitask Perceptron

In this section, we present our first family of online multitask classification algorithms, and prove a relative loss bound for the members of this family. This family includes algorithms for any global loss function defined through 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.

As previously noted, the Finite-Horizon Multitask Perceptron follows the general skeleton outlined in Figure 1. Given an absolute norm $\|\cdot\|$ and its dual $\|\cdot\|^*$, the multitask Perceptron sets $\tau_{t,j}$ in Figure 1 to

$$\tau_t \;=\; \operatorname*{argmax}_{\tau:\,\|\tau\|^*\leq C} \tau\cdot\ell_t \;\;, \tag{9}$$

where $C > 0$ is a constant which is specified later in this section. There may exist multiple solutions to the maximization problem above and at least one of these solutions induces a conservative update. In other words, we may assume that the solution to Equation (9) is such that $\tau_{t,j} = 0$ at every coordinate j where $\ell_{t,j} = 0$. To see that such a solution exists, take an arbitrary optimal solution τ and let $\hat{\tau}$ be defined by

$$\hat{\tau}_j \;=\; \begin{cases} \tau_j & \text{if } \ell_{t,j} \neq 0 \\ 0 & \text{if } \ell_{t,j} = 0. \end{cases}$$

Clearly, $\tau\cdot\ell_t = \hat{\tau}\cdot\ell_t$, whereas $\|\hat{\tau}\|^* \leq \|\tau\|^* \leq C$. If the optimization problem in Equation (9) has multiple solutions that induce conservative updates, assume that one is chosen arbitrarily.

An equivalent way of defining the solution to Equation (9) is by satisfying the equality $\tau_t\cdot\ell_t = C\|\ell_t\|$. To see this equivalence, note that the dual of $\|\cdot\|^*$ is defined by Equation (2) to be

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

However, since $\|\cdot\|^{**}$ is equivalent to $\|\cdot\|$ (see for instance Theorem 5.5.14 in Horn and Johnson, 1985), we get

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

Using the linearity of $\|\cdot\|^*$, we conclude that $\|\tau/C\|^* = \|\tau\|^*/C$ for any $C > 0$, and therefore the above becomes

$$C\|\ell\| \;=\; \max_{\tau:\,\|\tau\|^*\leq C} \tau\cdot\ell \;\;.$$

We conclude that

$$\tau_t\cdot\ell_t \;=\; C\|\ell_t\| \tag{10}$$

holds if and only if τ_t is a maximizer of Equation (9).

When the global loss function is a p-norm, the following definition of τ_t solves Equation (9):

$$\tau_{t,j} \;=\; \frac{C\ell_{t,j}^{p-1}}{\|\ell_t\|_p^{p-1}} \;\;. \tag{11}$$

When the global loss function is an r-max norm and π is a permutation such that $\ell_{t,\pi(1)} \geq \ldots \geq \ell_{t,\pi(k)}$, the following definition of τ_t is a solution to Equation (9):

$$\tau_{t,j} \;=\; \begin{cases} C & \text{if } \ell_{t,j} > 0 \text{ and } j \in \{\pi(1),\ldots,\pi(r)\} \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

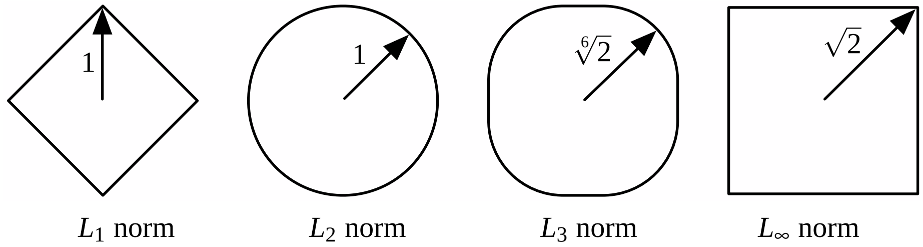

Figure 2: The *remoteness* of a norm is the longest Euclidean length of any vector contained in the norm's unit ball. The longest vector in each of the two-dimensional unit balls above is depicted with an arrow.

Note that when $r = k$, the r-max norm reduces to the L_1 norm and the above becomes the well-known update rule of the Perceptron algorithm (Rosenblatt, 1958; Novikoff, 1962). The correctness of the definitions in Equation (11) and Equation (12) can be easily verified by observing that $\|\tau_t\|^* \leq C$ and that $\tau_t \cdot \ell_t = C\|\ell_t\|$ in both cases.

Before proving a loss bound for the multitask Perceptron, we must introduce another important quantity. This quantity is the *remoteness* of a norm $\|\cdot\|$ defined on $\mathbb{R}^k$, and is defined to be

$$\rho(\|\cdot\|,k) \;=\; \max_{\mathbf{u}\in\mathbb{R}^k} \frac{\|\mathbf{u}\|_2}{\|\mathbf{u}\|} \;=\; \max_{\mathbf{u}\in\mathbb{R}^k:\|\mathbf{u}\|\leq 1} \|\mathbf{u}\|_2 \quad . \tag{13}$$

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 $\|\cdot\|$. This definition is visually depicted in Figure 2. As we show below, the remoteness of the dual norm, $\rho(\|\cdot\|^*,k)$, plays an important role in determining the difficulty of using $\|\cdot\|$ as the global loss function.

For concreteness, we now calculate the remoteness of the duals of p-norms and of r-max norms.

Lemma 2 *The remoteness of a p-norm $\|\cdot\|_q$ equals*

$$\rho(\|\cdot\|_q,k) \;=\; \begin{cases} 1 & \text{if } 1 \leq q \leq 2 \\ k^{\left(\frac{1}{2}-\frac{1}{q}\right)} & \text{if } 2 < q \end{cases} \quad .$$

Before proving the lemma, we note that if $\|\cdot\|_p$ is a p-norm and $\|\cdot\|_q$ is its dual, then we can combine Lemma 2 with the equality $q = \frac{p}{p-1}$ to obtain

$$\rho(\|\cdot\|_q,k) \;=\; \begin{cases} 1 & \text{if } 2 \leq p \\ k^{\left(\frac{1}{p}-\frac{1}{2}\right)} & \text{if } 1 \leq p < 2 \end{cases} \quad .$$

This equivalent form is better suited to our needs. The proof of Lemma 2 is given below.

Proof If $2 \leq p$ then $1 \leq q \leq 2$, and the monotonicity of the p-norms implies that $\|\mathbf{v}\|_q \geq \|\mathbf{v}\|_2$ for all $\mathbf{v} \in \mathbb{R}^k$. Therefore $\|\mathbf{v}\|_2/\|\mathbf{v}\|_q \leq 1$ for all $\mathbf{v} \in \mathbb{R}^k$ and thus $\rho(\|\cdot\|_q,k) \leq 1$. On the other hand, setting $\mathbf{v} = (1,0,\ldots,0)$, we get $\|\mathbf{v}\|_q = \|\mathbf{v}\|_2$ and therefore $\rho(\|\cdot\|_q,k) \geq 1$. Overall, we have shown that $\rho(\|\cdot\|_q,k) = 1$.

Turning to the case where $1 \leq p < 2$, we note that $q > 2$. Let $\mathbf{v}$ be an arbitrary vector in $\mathbb{R}^k$, and define $\mathbf{u} = (v_1^2,\ldots,v_k^2)$ and $\mathbf{w} = (1,\ldots,1)$. Noting that $\|\cdot\|_{\frac{q}{2}}$ and $\|\cdot\|_{\frac{q}{q-2}}$ are dual norms, we use Hölder's inequality to obtain

$$\mathbf{u}\cdot\mathbf{w} \;\leq\; \|\mathbf{u}\|_{\frac{q}{2}}\,\|\mathbf{w}\|_{\frac{q}{q-2}} \quad .$$

The left-hand side above equals $\|\mathbf{v}\|_2^2$, while the right-hand side above equals $\|\mathbf{v}\|_q^2 k^{1-\frac{2}{q}}$. Therefore, $\|\mathbf{v}\|_2^2/\|\mathbf{v}\|_q^2 \leq k^{1-\frac{2}{q}}$ and taking square-roots on both sides yields $\|\mathbf{v}\|_2/\|\mathbf{v}\|_q \leq k^{\frac{1}{2}-\frac{1}{q}}$. Since this inequality holds for all $\mathbf{v} \in \mathbb{R}^k$, we have shown that $\rho(\|\cdot\|_q,k) \leq k^{\frac{1}{2}-\frac{1}{q}}$. On the other hand, setting $\mathbf{v} = (1,\ldots,1)$, we get $\|\mathbf{v}\|_2 = k^{\frac{1}{2}-\frac{1}{q}}\|\mathbf{v}\|_q$. This proves that $\rho(\|\cdot\|_q,k) \geq k^{\frac{1}{2}-\frac{1}{q}}$, and therefore $\rho(\|\cdot\|_q,k) = k^{\frac{1}{2}-\frac{1}{q}}$. ■

Lemma 3 *Let $\|\cdot\|_{r\text{-max}}$ be a r-max norm and let $\|\cdot\|^*_{r\text{-max}}$ be its dual. The remoteness of $\|\cdot\|^*_{r\text{-max}}$ equals $\sqrt{r}$.*

Proof Using Equation (13), the remoteness of $\|\cdot\|^*_{r\text{-max}}$ is defined to be the maximum value of $\|\mathbf{u}\|_2$ subject to $\|\mathbf{u}\|^*_{r\text{-max}} \leq 1$. Recalling the definition of $\|\cdot\|^*_{r\text{-max}}$ from Equation (3), we can replace this constraint with two constraints $\|\mathbf{u}\|_1 \leq r$ and $\|\mathbf{u}\|_\infty \leq 1$. Moreover, since both the L_1 norm and the L_∞ norm are absolute norms, we can also assume that $\mathbf{u}$ resides in the non-negative orthant. Therefore, we have that $0 \leq u_j \leq 1$ for all $1 \leq j \leq k$. From this we conclude that $u_j^2 \leq u_j$ for all $1 \leq j \leq k$, and thus $\|\mathbf{u}\|_2^2 \leq \|\mathbf{u}\|_1 \leq r$. Hence, $\|\mathbf{u}\|_2 \leq \sqrt{r}$ and $\rho(\|\cdot\|^*_{r\text{-max}},k) \leq \sqrt{r}$. On the other hand, the vector

$$\mathbf{u} \;=\; \big(\overbrace{1,\ldots,1}^{r},\overbrace{0,\ldots,0}^{k-r}\big)$$

is contained in the unit ball of $\|\cdot\|^*_{r\text{-max}}$, and its Euclidean length is $\sqrt{r}$. Therefore, we also have that $\rho(\|\cdot\|^*_{r\text{-max}},k) \geq \sqrt{r}$, and overall we get $\rho(\|\cdot\|^*_{r\text{-max}},k) = \sqrt{r}$. ■

We are now ready to prove a loss bound for the Finite-Horizon Multitask Perceptron.

Theorem 4 *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 C be a positive constant and let $\|\cdot\|$ be an absolute norm. 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 incurred by $\mathbf{w}_j^\star$ on example $(\mathbf{x}_{t,j},y_{t,j})$ to be $\ell^\star_{t,j} = \big[1 - y_{t,j}\mathbf{w}_j^\star\cdot\mathbf{x}_{t,j}\big]_+$. If we present this sequence to the finite-horizon multitask Perceptron with the norm $\|\cdot\|$ and the aggressiveness parameter C, then,*

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

Proof The starting point of our analysis is Lemma 1. The choice of $\tau_{t,j}$ in Equation (9) is clearly bounded by $\|\tau_t\|^* \leq C$ and conservative. It is also non-negative, due to the fact that $\|\cdot\|^*$ is an absolute norm and that $\ell_{t,j} \geq 0$. Therefore, the definition of $\tau_{t,j}$ in Equation (9) meets the requirements of the lemma, and we have

$$\sum_{t=1}^{T}\sum_{j=1}^{k}\left(2\tau_{t,j}\ell_{t,j}-\tau_{t,j}^{2}\|\mathbf{x}_{t,j}\|_2^2\right)\ \leq\ \sum_{j=1}^{k}\|\mathbf{w}_j^{\star}\|_2^2\ +\ 2C\sum_{t=1}^{T}\|\ell_t^{\star}\|\ \ .$$

Using Equation (10), we rewrite the left-hand side of the above as

$$2C\sum_{t=1}^{T}\|\ell_t\|-\sum_{t=1}^{T}\sum_{j=1}^{k}\tau_{t,j}^{2}\|\mathbf{x}_{t,j}\|_2^2\ \ . \tag{14}$$

Using our assumption that $\|\mathbf{x}_{t,j}\|_2^2 \leq R^2$, we know that $\sum_{j=1}^{k}\tau_{t,j}^{2}\|\mathbf{x}_{t,j}\|_2^2 \leq (R\|\tau_t\|_2)^2$. Using the definition of remoteness, we can upper bound this term by $(R\|\tau_t\|^{*}\rho(\|\cdot\|^{*},k))^2$. Finally, using our upper bound on $\|\tau_t\|^{*}$ we can further bound this term by $R^2C^2\rho^2(\|\cdot\|^{*},k)$. Plugging this bound back into Equation (14) gives

$$2C\sum_{t=1}^{T}\|\ell_t\|\ -\ TR^2C^2\rho^2(\|\cdot\|^{*},k)\ \ .$$

Overall, we have shown that

$$2C\sum_{t=1}^{T}\|\ell_t\|\ -\ TR^2C^2\rho^2(\|\cdot\|^{*},k)\ \leq\ \sum_{j=1}^{k}\|\mathbf{w}_j^{\star}\|_2^2\ +\ 2C\sum_{t=1}^{T}\|\ell_t^{\star}\|\ \ .$$

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

In its current form, the bound in Theorem 4 may seem insignificant, since its right-most term grows linearly with the length of the input sequence, T. This term can be easily controlled by setting C to a value on the order of $1/\sqrt{T}$.

Corollary 5 *Under the assumptions of Theorem 4, if $C = 1/(\sqrt{T}R^2)$, then*

$$\sum_{t=1}^{T}\|\ell_t\|\ \leq\ \sum_{t=1}^{T}\|\ell_t^{\star}\|\ +\ \frac{\sqrt{T}}{2}\left(R^2\sum_{j=1}^{k}\|\mathbf{w}_j^{\star}\|_2^2\ +\ \rho^2(\|\cdot\|^{*},k)\right)\ \ .$$

This corollary bounds the global loss cumulated by our algorithm with the global loss obtained by any fixed set of hypotheses, plus a term which grows sub-linearly in T. The significance of this term depends on the magnitude of the constant

$$\frac{1}{2}\left(R^2\sum_{j=1}^{k}\|\mathbf{w}_j^{\star}\|_2^2\ +\ \rho^2(\|\cdot\|^{*},k)\right)\ \ .$$

Our algorithm uses C in its update procedure, and the value of C depends on $\sqrt{T}$. Therefore, the algorithm is a finite horizon algorithm.

Dividing both sides of the inequality in Corollary 5 by T, we see that the average global loss suffered by the multitask Perceptron is upper bounded by the average global loss of the best fixed hypothesis ensemble plus a term that diminishes with T. Using game-theoretic terminology, we can now say that the multitask Perceptron exhibits *no-regret* with respect to any global loss function defined by an absolute norm. The same cannot be said for the naive alternative of learning each task independently using a separate single-task Perceptron. We show this by presenting a simple counter-example. Specifically, we construct a concrete k-task problem with a specific global loss, an arbitrarily long input sequence $\{(\mathbf{x}_{t,j}, y_{t,j})\}_{1\leq t\leq T}^{1\leq j\leq k}$, and fixed weight vectors $\mathbf{u}_1,\ldots,\mathbf{u}_k$ to use for comparison. We then prove that

$$\frac{k+1}{2}\sum_{t=1}^{T}\|\ell^\star_t\|_\infty \;\leq\; \sum_{t=1}^{T}\|\hat{\ell}_t\|_\infty \;\;, \tag{15}$$

where $\hat{\ell}_t$ is the vector of individual losses of the k independent single-task Perceptrons, and, as before, $\ell^\star_t$ is the vector of individual losses of $\mathbf{u}_1,\ldots,\mathbf{u}_k$ respectively. This example demonstrates that a claim along the lines of Corollary 5 cannot be proven for the set of independent single-task Perceptrons.

First, we would like to emphasize that we are considering a version of the single-task Perceptron that updates its hypothesis whenever it suffers a positive hinge-loss, and not only when it makes a prediction mistake. Moreover, when an update is performed, the algorithm defines $\mathbf{w}_{t+1} = \mathbf{w}_t + Cy_t\mathbf{x}_t$, where C is a predefined constant. This version of the Perceptron is sometimes called the *aggressive Perceptron*. If we were to use the simplest version of the Perceptron, which updates its hypothesis only when a prediction mistake occurs, then finding a counter-example that achieves Equation (15) would be trivial, without even using the distinction between single-task and multitask Perceptron learning.

Also, we can assume without loss of generality that $1/C = o(T)$, since otherwise, even in the case $k = 1$, simply repeating the same example over and over provides a counterexample.

Moving on to the counter-example itself, assume that our global loss is defined by the L_∞ norm. Let k be at least 2, assume that the instances of all k problems are two dimensional vectors, and set $\mathbf{u}_1 = \ldots = \mathbf{u}_k = (1,1)$. Each of the single-task Perceptrons initializes its hypothesis to $(0,0)$. Assume that all of the labels in the input sequence are positive labels. For $t = 0$, we set $\mathbf{x}_{1,1} = \ldots = \mathbf{x}_{1,k} = (1,0)$. Each one of the independent Perceptrons suffers a positive individual loss and updates its weight vector to $(C,0)$. We continue presenting the same example for $\lceil 1/C\rceil - 1$ additional rounds, which is precisely when all k weight vectors of the Perceptrons become equal to $(\alpha,0)$, with $\alpha \geq 1$. For instance, if $C = O(1/\sqrt{T})$ then the vector $(1,0)$ is presented $O(\sqrt{T})$ times. Meanwhile, the fixed weight vectors $\mathbf{u}_1,\ldots,\mathbf{u}_k$ suffer no loss at all.

Define $t_0 = \lceil 1/C\rceil$, and note that the index of the next online round is t_0+1. For each t in $t_0+1,\ldots,t_0+k$, we set $\mathbf{x}_{t,t-t_0}$ to $(0,1)$ and $\mathbf{x}_{t,j}$ to $(1,0)$ for all $j \neq t-t_0$. On round t, the $(t-t_0)$'th Perceptron, whose weight vector is $(\alpha,0)$, suffers an individual loss of 1 and updates its weight vector to (α,C). The remaining $k-1$ Perceptrons suffer no individual loss and do not modify their weight vectors. Consequently, $\|\hat{\ell}_t\|_\infty = 1$ on each of these rounds. Once again, the fixed vectors $\mathbf{u}_1,\ldots,\mathbf{u}_k$ suffer no loss at all. On round $t = t_0+k+1$, we set $\mathbf{x}_{t,1} = \ldots = \mathbf{x}_{t,k} = (0,-1)$. As a result, each of the Perceptrons suffers a hinge loss of $1+C$ and updates its weight vector back to $(\alpha,0)$. Since C is positive, we get $\|\hat{\ell}_t\|_\infty \geq 1$. Meanwhile, $\|\ell^\star_t\|_\infty = 2$. We now have that

$$\sum_{t=t_0+1}^{t_0+k+1}\|\hat{\ell}_t\|_\infty \;\geq\; k+1 \quad\text{and}\quad \sum_{t=t_0+1}^{t_0+k+1}\|\ell^\star_t\|_\infty \;=\; 2 \;\;.$$

Furthermore, the weight vectors of the k single-task Perceptrons have returned to their values at the end of round t_0. Therefore, by repeating the input sequence from round t_0+1 to round t_0+k+1 over and over again, we obtain Equation (15).

This concludes the presentation of the counter-example thus showing that a set of independent single-task Perceptrons does not attain no-regret with respect to the L_∞ norm global loss. Similar constructions can be given for other global loss functions. The exception is the L_1 norm, which naturally reduces the multitask Perceptron to k independent single-task Perceptrons.

4. An Extension to the Infinite Horizon Setting

In the previous section, we devised an algorithm which relied on prior knowledge of T, the input sequence length. In this section, we adapt the update procedure from the previous section to the infinite horizon setting, where T is not known in advance. Moreover, the bound we prove in this section holds simultaneously for every prefix of the input sequence. This generalization comes at a price; we can only prove an upper bound on $\sum_t \min\{\ell_t, \ell_t^2\}$, a quantity similar to the cumulative global loss, but not the global loss per se.

To motivate our infinite-horizon algorithm, we take a closer look at the analysis of the finite-horizon algorithm. In the proof of Theorem 4, 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 $2C\|\ell_t\| - R^2C^2\rho^2(\|\cdot\|^*, k)$. The first term in this lower bound is proportional to the global loss suffered on round t, and the second term is a constant. When $\|\ell_t\|$ is smaller than this constant, our lower bound becomes negative. This suggests that the update step-size applied by the finite-horizon Perceptron may have been too large, and that the update step may have overshot its target. As a result, the new hypothesis may be inferior to the previous one. Nevertheless, over the course of T rounds, our positive progress is guaranteed to overshadow our negative progress, and thus we are able to prove Theorem 4. However, if we are interested in a bound which holds for every prefix of the input sequence, we must ensure that every individual update makes positive progress. Concretely, 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 guaranteed to be non-negative. The vector τ_t remains in the same direction as before, but by setting its length more carefully, we enforce an update step-size which is never excessively large.

We use ρ to abbreviate $\rho(\|\cdot\|^*, k)$ throughout this section. We replace the definition of τ_t in Equation (9) with the following definition,

$$\tau_t \;=\; \underset{\tau:\, \|\tau\|^* \le \min\left\{C, \frac{\|\ell_t\|}{R^2\rho^2}\right\}}{\operatorname{argmax}} \; \tau \cdot \ell_t \;\;, \tag{16}$$

where $C > 0$ is a user defined parameter and $R > 0$ is an upper bound on $\|\mathbf{x}_{t,j}\|_2$ for all $1 \le t \le T$ and all $1 \le j \le k$. As in the previous section, we assume that $\tau_{t,j} = 0$ whenever $\ell_{t,j} = 0$. As in Equation (10), the solution to Equation (16) can be equivalently defined by the equation

$$\tau_t \cdot \ell_t \;=\; \min\left\{C, \frac{\|\ell_t\|}{R^2\rho^2}\right\} \|\ell_t\| \;\;. \tag{17}$$

When the global loss function is a p-norm, the following definition of τ_t solves Equation (16):

$$\tau_{t,j} \;=\; \begin{cases} \frac{\ell_{t,j}^{p-1}}{R^2\rho^2\|\ell_t\|_p^{p-2}} & \text{if } \|\ell_t\|_p \le R^2C\rho^2 \\ \frac{C\ell_{t,j}^{p-1}}{\|\ell_t\|_p^{p-1}} & \text{if } \|\ell_t\|_p > R^2C\rho^2 . \end{cases}$$

When 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 the following definition of τ_t is a solution to Equation (16):

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

The correctness of both definitions of $\tau_{t,j}$ given above can be verified by observing that $\|\tau_t\|^* \leq \min\{C, \frac{\|\ell_t\|}{R^2\rho^2}\}$ and that $\tau_t \cdot \ell_t = \min\{C, \frac{\|\ell_t\|}{R^2\rho^2}\}\|\ell_t\|$ in both cases. We now turn to proving an infinite-horizon cumulative loss bound for our algorithm.

Theorem 6 *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 C be a positive constant, let $\|\cdot\|$ be an absolute norm, and let ρ be an abbreviation for $\rho(\|\cdot\|^*, k)$. 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 explicit multitask algorithm with the norm $\|\cdot\|$ and the aggressiveness parameter C, then for every T*

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

Proof The starting point of our analysis is again Lemma 1. The choice of $\tau_{t,j}$ in Equation (16) is clearly bounded by $\|\tau_t\|^* \leq C$ and conservative. It is also non-negative, due to the fact that $\|\cdot\|^*$ is absolute and that $\ell_{t,j} \geq 0$. Therefore, $\tau_{t,j}$ meets the requirements of Lemma 1, and we have

$$\sum_{t=1}^{T}\sum_{j=1}^{k}\left(2\tau_{t,j}\ell_{t,j} - \tau_{t,j}^2\|\mathbf{x}_{t,j}\|_2^2\right) \;\leq\; \sum_{j=1}^{k}\|\mathbf{w}_j^\star\|_2^2 \;+\; 2C\sum_{t=1}^{T}\|\ell_t^\star\| \ \ . \qquad (18)$$

We now prove our theorem by lower-bounding the left hand side of Equation (18) above. We analyze two different cases. First, if $\|\ell_t\| \leq R^2C\rho^2$ then $\min\{C, \|\ell_t\|/(R^2\rho^2)\} = \|\ell_t\|/(R^2\rho^2)$. Together with Equation (17), this gives

$$2\sum_{j=1}^{k}\tau_{t,j}\ell_{t,j} \;=\; 2\|\tau_t\|^*\,\|\ell_t\| \;=\; 2\frac{\|\ell_t\|^2}{R^2\rho^2} \ \ . \qquad (19)$$

On the other hand, $\sum_{j=1}^{k}\tau_{t,j}^2\|\mathbf{x}_{t,j}\|_2^2$ can be bounded by $\|\tau_t\|_2^2 R^2$. Using the definition of remoteness, we bound this term by $(\|\tau_t\|^*)^2 R^2\rho^2$. Using the fact that, $\|\tau_t\|^* \leq \|\ell_t\|/(R^2\rho^2)$, we bound this term by $\|\ell_t\|^2/(R^2\rho^2)$. Overall, we have shown that

$$\sum_{j=1}^{k}\tau_{t,j}^2\|\mathbf{x}_{t,j}\|_2^2 \;\leq\; \frac{\|\ell_t\|^2}{R^2\rho^2} \ \ .$$

Subtracting both sides of the above inequality from the respective sides of Equation (19) gives

$$\frac{\|\ell_t\|^2}{R^2\rho^2} \;\leq\; \sum_{j=1}^{k}\left(2\tau_{t,j}\ell_{t,j} - \tau_{t,j}^2\|\mathbf{x}_{t,j}\|_2^2\right) \ \ . \qquad (20)$$

Moving on to the second case, if $\|\ell_t\| > R^2 C \rho^2$ then $\min\{C, \|\ell_t\|/(R^2\rho^2)\} = C$. Using Equation (17), we have that

$$2\sum_{j=1}^{k} \tau_{t,j}\ell_{t,j} \;=\; 2\|\tau_t\|^* \, \|\ell_t\| \;=\; 2C\|\ell_t\| \;\; . \tag{21}$$

As before, we can upper bound $\sum_{j=1}^{k} \tau_{t,j}^2 \|\mathbf{x}_{t,j}\|_2^2$ by $(\|\tau_t\|^*)^2 R^2 \rho^2$. Using the fact that, $\|\tau_t\|^* \leq C$ we can bound this term by $C^2 R^2 \rho^2$. Finally, using our assumption that $\|\ell_t\| > R^2 C \rho^2$, we conclude that

$$\sum_{j=1}^{k} \tau_{t,j}^2 \|\mathbf{x}_{t,j}\|_2^2 \;<\; C\|\ell_t\| \;\; .$$

Subtracting both sides of the above inequality from the respective sides of Equation (21) gives

$$C\|\ell_t\| \;\leq\; \sum_{j=1}^{k} \left(2\tau_{t,j}\ell_{t,j} - \tau_{t,j}^2 \|\mathbf{x}_{t,j}\|_2^2\right) \;\; . \tag{22}$$

Comparing the upper bound in Equation (18) with the lower bounds in Equation (20) and Equation (22) proves the theorem. ■

Corollary 7 *Under the assumptions of Theorem 6, if C is set to be $1/(R^2\rho^2)$ then for every $T' \leq T$ it holds that,*

$$\sum_{t=1}^{T'} \min\left\{\|\ell_t\|^2, \|\ell_t\|\right\} \;\leq\; 2\sum_{t=1}^{T'} \|\ell_t^\star\| \;+\; R^2\rho^2 \sum_{j=1}^{k} \|\mathbf{w}_j^\star\|_2^2 \;\; .$$

As noted at the beginning of this section, we do not obtain a cumulative loss bound per se, but rather at a bound on $\sum_t \min\{\ell_t, \ell_t^2\}$. However, this bound holds simultaneously for every prefix of the input sequence, and the algorithm does not rely on knowledge of the input sequence length.

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. Although the implementation of specific members of this family may be more involved than the implementation of the multitask Perceptron, we recommend using this family of algorithms in practice. On every round, the set of hypotheses is updated according to the update rule:

$$\begin{aligned}\{\mathbf{w}_{t+1,1}, \ldots, \mathbf{w}_{t+1,k}\} \;\;=\;\; & \operatorname*{argmin}_{\mathbf{w}_1,\ldots,\mathbf{w}_k} \; \frac{1}{2}\sum_{j=1}^{k} \|\mathbf{w}_j - \mathbf{w}_{t,j}\|_2^2 + C\|\xi\| \\ & \text{s.t.} \;\; \forall j \quad \mathbf{w}_j \cdot \mathbf{x}_{t,j} \geq 1 - \xi_j \quad \text{and} \quad \xi_j \geq 0.\end{aligned} \tag{23}$$

This optimization problem captures the fundamental tradeoff inherent to online learning. On one hand, the term $\sum_{j=1}^{k} \|\mathbf{w}_j - \mathbf{w}_{t,j}\|_2^2$ in the objective function above keeps the new set of hypotheses close to the current set of hypotheses, so as to retain the information learned on previous rounds. On

```
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 sign$(\mathbf{w}_{t,j} \cdot \mathbf{x}_{t,j})$                [$1 \le j \le k$]
  • receive $y_{t,1},\ldots,y_{t,k}$
  • suffer loss $\ell_{t,j} = \left[1 - y_{t,j}\mathbf{w}_{t,j} \cdot \mathbf{x}_{t,j}\right]_+$        [$1 \le j \le k$]
  • update:
      $\{\mathbf{w}_{t+1,1},\ldots,\mathbf{w}_{t+1,k}\} = \operatorname*{argmin}_{\mathbf{w}_1,\ldots,\mathbf{w}_k} \frac{1}{2}\sum_{j=1}^{k} \|\mathbf{w}_j - \mathbf{w}_{t,j}\|_2^2 + C\|\xi\|$
                                  s.t. $\forall j \;\; \mathbf{w}_j \cdot \mathbf{x}_{t,j} \ge 1 - \xi_j$ and $\xi_j \ge 0$
```

Figure 3: The implicit update algorithm

the other hand, the term $\|\xi\|$ in the objective function, together with the constraints on ξ_j, forces the algorithm to make progress using the new examples obtained on this round. Different choices of the global loss function lead to different definitions of this progress. The pseudo-code of the implicit update algorithm is presented in Figure 3.

Our first task is to show that this update procedure follows the skeleton outlined in Figure 1, and satisfies the requirements of Lemma 1. We do so by finding the dual of the optimization problem given in Equation (23).

Lemma 8 *Let $\|\cdot\|$ be a norm and let $\|\cdot\|^*$ be its dual. Then the online update defined in Equation (23) is 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*

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

Moreover, this update is conservative.

Proof The update step in Equation (23) sets the vectors $\mathbf{w}_{t+1,1},\ldots,\mathbf{w}_{t+1,k}$ to be the solution to the following constrained minimization problem:

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

We begin by using 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 (Boyd and Vandenberghe, 2004) holds, and the above problem is equivalent to

$$\max_{\tau \geq 0} \min_{\mathbf{w}_1,\ldots,\mathbf{w}_k,\xi \geq 0} \mathcal{L}(\tau, \mathbf{w}_1, \ldots, \mathbf{w}_k, \xi) \ \ ,$$

where $\mathcal{L}(\tau, \mathbf{w}_1, \ldots, \mathbf{w}_k, \xi)$ is defined as follows:

$$\frac{1}{2} \sum_{j=1}^{k} \|\mathbf{w}_j - \mathbf{w}_{t,j}\|_2^2 + C\|\xi\| + \sum_{j=1}^{k} \tau_j \left(1 - y_{t,j} \mathbf{w}_j \cdot \mathbf{x}_{t,j} - \xi_j\right) \ \ .$$

We can rewrite $\mathcal{L}$ as the sum of two terms, the first a function of τ and $\mathbf{w}_1, \ldots, \mathbf{w}_k$ (denoted $\mathcal{L}_1$) and the second a function of τ and $\xi_1, \ldots, \xi_k$ (denoted $\mathcal{L}_2$),

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

Using the notation defined above, our optimization problem becomes,

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

For any choice of τ, $\mathcal{L}_1$ 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. Namely,

$$\forall j, l \quad 0 \ = \ \frac{\partial \mathcal{L}_1}{\partial w_{j,l}} \ = \ w_{j,l} - w_{t,j,l} - \tau_j y_{t,j} x_{t,j,l} \ \ .$$

from the above 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 next step is to show that the update is conservative. If $\ell_{t,j} = 0$ then setting $\mathbf{w}_j = \mathbf{w}_{t,j}$ satisfies the constraint $y_{t,j} \mathbf{w}_j \cdot \mathbf{x}_{t,j} \geq 1 - \xi_j$ with any choice of $\xi_j \geq 0$. Since choosing $\mathbf{w}_j = \mathbf{w}_{t,j}$ minimizes $\|\mathbf{w}_t - \mathbf{w}_{t,j}\|_2^2$ and does not restrict our choice of any other variable, then it is optimal. The relation between $\mathbf{w}_j$ and τ_j now implies that $\tau_j = 0$ whenever $\ell_{t,j} = 0$).

Plugging our expression for $\mathbf{w}_j$ into $\mathcal{L}_1$, we have that

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

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

$$\tau_t \ = \ \underset{\tau \geq 0}{\operatorname{argmax}} \left(\sum_{j=1}^{k} \left(\tau_j \ell_{t,j} - \frac{1}{2} \tau_j^2 \|\mathbf{x}_{t,j}\| \right) \ + \ \min_{\xi \geq 0} \mathcal{L}_2(\tau, \xi) \right) \ \ .$$

We finally turn our attention to $\mathcal{L}_2$ and abbreviate $B(\tau) = \min_{\xi \geq 0} \mathcal{L}_2(\tau, \xi)$. We now claim that B is a barrier function for the constraint $\|\tau\|^* \leq C$, namely

$$B(\tau) \ = \ \begin{cases} 0 & \text{if } \|\tau\|^* \leq C \\ -\infty & \text{if } \|\tau\|^* > C \end{cases} \ \ .$$

To see why this is true, recall that $\|\tau\|^*$ is defined to be

$$\|\tau\|^* \;=\; \max_{\varepsilon\in\mathbb{R}^k} \frac{\sum_{j=1}^k \tau_j \varepsilon_j}{\|\varepsilon\|} \;\;.$$

First, let us consider the case where $\|\tau\|^* > C$. In this case there exists a vector $\bar{\varepsilon}$ for which

$$\sum_{j=1}^k \tau_j \bar{\varepsilon}_j - C\|\bar{\varepsilon}\| \;>\; 0 \;\;.$$

Denote the left hand side of the above by δ. We can assume w.l.o.g. that all the components of $\bar{\varepsilon}$ are non-negative since $\tau \geq 0$. For any $c \geq 0$, we now have that

$$B(\tau) \;=\; \min_{\xi\geq 0} \mathcal{L}_2(\tau,\xi) \;\leq\; \mathcal{L}_2(\tau, c\bar{\varepsilon}) \;=\; -c\delta \;\;.$$

Therefore, by taking c to infinity we get that $B(\tau) = -\infty$.

Turning to the case $\|\tau\|^* \leq C$, we have that $\sum_{j=1}^k \tau_j \xi_j \leq C\|\xi\|$ for any choice of ξ, or in other words, $\min_{\xi\geq 0} \mathcal{L}_2(\tau,\xi) \geq 0$. However, this lower bound is attainable by setting $\xi = 0$. We conclude that if $\|\tau\|^* \leq C$ then $B(\tau) = 0$. The original optimization problem has reduced to the form

$$\tau_t \;=\; \operatorname*{argmax}_{\tau\geq 0} \left(\sum_{j=1}^k \left(\tau_j \ell_{t,j} - \frac{1}{2}\tau_j^2 \|\mathbf{x}_{t,j}\| \right) \;+\; B(\tau) \right) \;\;.$$

Clearly, the above is maximized in the domain where $B(\tau) = 0$. Therefore, we replace the function B with the constraint $\|\tau\|^* \leq C$, and get

$$\tau_t \;=\; \operatorname*{argmax}_{\tau\geq 0 \,:\, \|\tau\|^* \leq C} \;\; \sum_{j=1}^k \left(\tau_j \ell_{t,j} - \frac{1}{2}\tau_j^2 \|\mathbf{x}_{t,j}\| \right) \;\;.$$

■

Lemma 5 proves that the implicit update essentially finds the value of τ_t that maximizes the left-hand side of the bound in Lemma 1. This choice of τ_t produces the tightest loss bounds that can be derived from Lemma 1. In this sense, the implicit update algorithm takes full advantage of our proof technique. An immediate consequence of this observation is that the loss bounds of the multitask Perceptron also hold for the implicit algorithm. More precisely, the bound in Theorem 4 (and Corollary 5) holds not only for the multitask Perceptron, but also for the implicit update algorithm. Equivalently, it can be shown that the bound in Theorem 6 (and Corollary 7) also holds for the implicit update algorithm. We prove this formally below.

Theorem 9 *The bound in Theorem 4 also holds for the implicit update algorithm.*

Proof Let $\tau'_{t,j}$ denote the weights defined by the multitask Perceptron in Equation (9) and let $\tau_{t,j}$ denote the weights assigned by the implicit update algorithm. In the proof of Theorem 4, we showed that,

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

According to Lemma 8, the weights $\tau_{t,j}$ maximize,

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

subject to the constraints $\|\tau_t\|^* \leq C$ and $\tau_{t,j}\geq 0$. Since the weights $\tau'_{t,j}$ also satisfy these constraints, it holds that,

$$\sum_{j=1}^{k}\left(2\tau'_{t,j}\ell_{t,j}-\tau'^{2}_{t,j}\|\mathbf{x}_{t,j}\|_2^2\right) \;\leq\; \sum_{j=1}^{k}\left(2\tau_{t,j}\ell_{t,j}-\tau_{t,j}^{2}\|\mathbf{x}_{t,j}\|_2^2\right) \quad .$$

Therefore, we conclude that

$$2C\|\ell_t\|-R^2C^2\rho^2 \;\leq\; \sum_{j=1}^{k}\left(2\tau_{t,j}\ell_{t,j}-\tau_{t,j}^{2}\|\mathbf{x}_{t,j}\|_2^2\right) \quad . \tag{24}$$

Since $\tau_{t,j}$ is bounded, non-negative, and conservative (due to Lemma 8), the right-hand side of the above inequality is upper-bounded by Lemma 1. Comparing the bound in Equation (24) with the bound in Lemma 1 proves the theorem. ■

In the remainder of this paper, we present efficient algorithms which solve the optimization problem in Equation (23) for different choices of the global loss function.

6. Solving the Implicit Update for the L_2 Norm

Consider the implicit update with the L_2 norm, namely we are trying to solve

$$\tau_t \;=\; \underset{\tau\geq 0\,:\,\|\tau\|_2\leq C}{\operatorname{argmax}} \;\sum_{j=1}^{k}\left(\tau_j\ell_{t,j}-\frac{1}{2}\tau_j^2\|\mathbf{x}_{t,j}\|\right) \quad .$$

The Lagrangian of this optimization problem is

$$\mathcal{L} \;=\; \sum_{j=1}^{k}\left(2\tau_{t,j}\ell_{t,j}-\tau_{t,j}^{2}\|\mathbf{x}_{t,j}\|_2^2\right)-\theta\left(\sum_{j=1}^{k}\tau_{t,j}^2-C^2\right) \quad ,$$

where θ is a non-negative Lagrange multiplier. The derivative of $\mathcal{L}$ with respect to each $\tau_{t,j}$ is, $2\ell_{t,j}-2\tau_{t,j}\|\mathbf{x}_{t,j}\|_2^2-2\theta\tau_{t,j}$. Setting this derivative to zero, we get

$$\tau_{t,j} \;=\; \frac{\ell_{t,j}}{\|\mathbf{x}_{t,j}\|_2^2+\theta} \quad .$$

The optimum of the *unconstrained* problem is attained by choosing $\tau_{t,j}=\frac{\ell_{t,j}}{\|\mathbf{x}_{t,j}\|_2^2}$ for each j. If, for this choice of τ_t, the constraint $\sum_{j=1}^{k}\tau_{t,j}^2\leq C^2$ does not hold, then θ must be greater than zero. The KKT complementarity condition implies that in this case the constraint is binding, namely $\sum_{j=1}^{k}\tau_{t,j}^2=C^2$. In order to find θ, we must now solve the following equation:

$$\sum_{j=1}^{k}\left(\frac{\ell_{t,j}}{\|\mathbf{x}_{t,j}\|_2^2+\theta}\right)^2=C^2 \quad . \tag{25}$$

The left hand side of the above is monotonically decreasing in θ. We also know that $\theta > 0$. Moreover, setting

$$\theta \;=\; \frac{\sqrt{k}\|\ell_t\|_\infty}{C}$$

in the left-hand side of Equation (25) yields a value which is at least C^2, and therefore we conclude that $\theta \leq \frac{\sqrt{k}\|\ell_t\|_\infty}{C}$. These properties enable us to easily find θ using binary search.

In the special case where the norms of all the instances are equal, namely $\|\mathbf{x}_{t,1}\|_2^2 = \ldots = \|\mathbf{x}_{t,k}\|_2^2 = R^2$, Equation (25) gives $\theta \;=\; \frac{\|\ell_t\|_2}{C} - R^2$, and therefore $\tau_{t,j} = C\ell_{t,j}/\|\ell_t\|_2$. The general expression for $\tau_{t,j}$ in this case becomes

$$\tau_{t,j} \;=\; \begin{cases} \frac{\ell_{t,j}}{R^2} & \text{if } \|\ell_t\|_2 \;\leq\; R^2C \\ \frac{C\ell_{t,j}}{\|\ell_t\|_2} & \text{otherwise} \end{cases} \qquad .$$

Note that the above coincides with the definition of τ_t given by the Infinite Horizon Multitask Perceptron for the L_2 norm, as defined in Section 4.

7. Solving the Implicit Update for *r*-max Norms

We now present an efficient procedure for calculating the update in Equation (23), in the case where the norm being used is the *r*-max norm. Lemma 8, together with (3), tells us that the update can be calculated by solving the following constrained optimization problem:

$$\begin{aligned} \tau_t \quad &= \quad \operatorname*{argmax}_{\tau} \sum_{j=1}^{k} \left(2\tau_j\ell_{t,j} - \tau_j^2\|\mathbf{x}_{t,j}\|_2^2\right) && (26) \\ &\text{s.t.} \quad \sum_{j=1}^{k}\tau_j \leq Cr \;,\; \forall j \;\; \tau_j \leq C \;,\; \forall j \;\; \tau_j \geq 0 \;\; . \end{aligned}$$

After dividing the objective function by 2, the Lagrangian of this optimization problem is

$$\sum_{j=1}^{k}\left(\tau_j\ell_{t,j} - \frac{1}{2}\tau_j^2\|\mathbf{x}_{t,j}\|_2^2\right) + \theta\left(Cr - \sum_{j=1}^{k}\tau_j\right) + \sum_{j=1}^{k}\lambda_j(C-\tau_j) + \sum_{j=1}^{k}\beta_j\tau_j \;\; ,$$

where θ, the β_j's and the λ_j's are non-negative Lagrange multipliers. The derivative of $\mathcal{L}$ with respect to each τ_j is, $\ell_{t,j} - \tau_j\|\mathbf{x}_{t,j}\|_2^2 - \theta - \lambda_j + \beta_j$. All of these partial derivatives must equal zero at the optimum, and therefore

$$\forall\, 1 \leq j \leq k \qquad \tau_j = \frac{\ell_{t,j} - \theta - \lambda_j + \beta_j}{\|\mathbf{x}_{t,j}\|_2^2} \quad . \tag{27}$$

The KKT complementarity condition states that the following equalities hold at the optimum:

$$\forall\, 1 \leq j \leq k \qquad \lambda_j(C - \tau_j) = 0 \qquad \text{and} \qquad \beta_j\tau_j = 0 \;. \tag{28}$$

We consider three different cases:

1. Assume that $\ell_{t,j} - \theta < 0$. Since both τ_j and λ_j must be non-negative, then from the definition of τ_j in Equation (27) we learn that β_j must be at least $\theta - \ell_{t,j}$. In other words, β_j is positive. Referring to the right-hand side of Equation (28), we conclude that $\tau_j = 0$.

2. Assume that $0 \leq \ell_{t,j} - \theta \leq C\|\mathbf{x}_{t,j}\|_2^2$. Summing the two equalities in Equation (28) and plugging in the definition of τ_j from Equation (27) results in,

$$\lambda_j \left(C - \frac{\ell_{t,j} - \theta}{\|\mathbf{x}_{t,j}\|_2^2} \right) + \beta_j \left(\frac{\ell_{t,j} - \theta}{\|\mathbf{x}_{t,j}\|_2^2} \right) + \frac{(\beta_j - \lambda_j)^2}{\|\mathbf{x}_{t,j}\|_2^2} \;=\; 0 \quad . \tag{29}$$

   Using our assumption that $\ell_{t,j} - \theta \geq 0$, along with the requirement that $\beta_j \geq 0$, gives us that $\beta(\ell_{t,j} - \theta)/\|\mathbf{x}_{t,j}\|_2^2 \geq 0$. Equivalently, using our assumption that $\ell_{t,j} - \theta \leq C\|\mathbf{x}_{t,j}\|_2^2$ along with the requirement that $\lambda_j \geq 0$ results in $\lambda\big(C - (\ell_{t,j} + \theta)/\|\mathbf{x}_{t,j}\|_2^2\big) \geq 0$. Plugging the last two inequalities back into Equation (29) gives, $(\beta_j - \lambda_j)^2/\|\mathbf{x}_{t,j}\|_2^2 \leq 0$. The only way that this inequality can hold is if $(\beta_j - \lambda_j) = 0$. Thus, the definition of τ_j in Equation (27) reduces to $\tau_j \;=\; \frac{\ell_{t,j} - \theta}{\|\mathbf{x}_{t,j}\|_2^2}$.

3. Finally, assume that $\ell_{t,j} - \theta > C\|\mathbf{x}_{t,j}\|_2^2$. Since $\tau_j \leq \|\tau\|_\infty \leq C$ and $\beta_j \geq 0$, then from Equation (27) we conclude that λ_j is at least $\ell_{t,j} - \theta - C\|\mathbf{x}_{t,j}\|_2^2$. In other words, λ_j is positive. Referring to the left-hand side of Equation (28), we conclude that $(C - \tau_j) = 0$, and $\tau_j = C$.

Overall, we have shown that there exists some $\theta \geq 0$ such that the optimal update weights take the form

$$\tau_{t,j} \;=\; \begin{cases} 0 & \text{if } \ell_{t,j} - \theta < 0 \\ \frac{\ell_{t,j} - \theta}{\|\mathbf{x}_{t,j}\|_2^2} & \text{if } 0 \leq \ell_{t,j} - \theta \leq C\|\mathbf{x}_{t,j}\|_2^2 \\ C & \text{if } C\|\mathbf{x}_{t,j}\|_2^2 < \ell_{t,j} - \theta \end{cases} \quad . \tag{30}$$

That is, if the individual loss of task j is smaller than θ then no update is applied to the respective classifier. If the loss is moderate then the size of the update step is proportional to the loss attained, and inverse proportional to the squared norm of the respective instance. In any case, the size of the update step cannot exceed the fixed upper limit C.

We are thus left with the problem of finding the value of θ in Equation (30) which yields the update weights that maximize Equation (26). We denote this value by $\theta^\star$. First note that if we lift the constraint $\sum_{j=1}^k \tau_{t,j} \leq rC$ then the maximum of Equation (26) is obtained by setting $\tau_{t,j} = \min\{\ell_{t,j}/\|\mathbf{x}_{t,j}\|_2^2,\, C\}$ for all j, which is equivalent to setting $\theta = 0$ in Equation (30). Therefore, if

$$\sum_{j=1}^{k} \min\left\{ \frac{\ell_{t,j}}{\|\mathbf{x}_{t,j}\|_2^2}, C \right\} \;\leq\; rC \;\;,$$

the solution to Equation (26) is $\tau_{t,j} = \min\{\ell_{t,j}/\|\mathbf{x}_{t,j}\|_2^2,\, C\}$ for all j. Thus, we can focus our attention on the case where

$$\sum_{j=1}^{k} \min\left\{ \frac{\ell_{t,j}}{\|\mathbf{x}_{t,j}\|_2^2}, C \right\} \;>\; rC \;\;.$$

In this case, $\theta^\star$ must be non-zero in order for the constraint $\sum_{j=1}^k \tau_j \leq rC$ to hold. Once again using the KKT complementarity condition, it follows that $\sum_{j=1}^k \tau_{t,j} = rC$. Now, for every value of θ, define the following two sets of indices:

$$\Psi(\theta) \;=\; \{1 \leq j \leq k \;:\; 0 < \ell_{t,j} - \theta\} \;\;,$$

and

$$\Phi(\theta) \;=\; \{1 \le j \le k \;:\; C\|\mathbf{x}_{t,j}\|_2^2 < \ell_{t,j} - \theta\} \;\; .$$

Let Ψ and Φ denote the sets $\Psi(\theta^\star)$ and $\Phi(\theta^\star)$ respectively. The semantics of Ψ and Φ are readily available from Equation (30): the set Ψ includes all indices j for which $\tau_j > 0$ in the optimal solution, while Φ includes all indices j for which τ_j is clipped at C in the optimal solution. If we know the value of $\theta^\star$, we can easily obtain the sets Ψ and Φ from their definitions above. However, the converse is also true: if we are able to find the sets Ψ and Φ directly then we can use them to calculate the exact value of $\theta^\star$. Assuming we know Ψ and Φ, and using the fact that $\sum_{j=1}^k \tau_j = rC$, we get

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

Solving the above for $\theta^\star$ gives

$$\theta^\star \;=\; \frac{\sum_{j \in \Psi \setminus \Phi} \frac{\ell_{t,j}}{\|\mathbf{x}_{t,j}\|_2^2} - rC + \sum_{j \in \Phi} C}{\sum_{j \in \Psi \setminus \Phi} \frac{1}{\|\mathbf{x}_{t,j}\|_2^2}} \;\; . \qquad (31)$$

We have thus reduced the optimization problem in Equation (26) to the problem of finding the sets Ψ and Φ. Once we find Ψ and Φ, we can easily calculate $\theta^\star$ using Equation (31) and then obtain τ_t using Equation (30). Luckily, Ψ and Φ are subsets of $\{1,\ldots,k\}$ and can only be defined in a finite number of ways. A straightforward and excessively inefficient solution is to enumerate over all possible subsets of $\{1,\ldots,k\}$ as candidates for Ψ and Φ, for each pair of candidate sets to compute the corresponding values of θ and τ using Equation (31) and Equation (30) respectively and then check if the obtained solution is consistent with our constraints ($\theta \ge 0$, $\sum_j \tau_j = rC$ and $0 \le \tau_j \le C$). Of the candidates that turn out to be consistent, we choose the one which maximizes the objective function in Equation (26). This approach is clearly infeasible even for reasonably small values of k. We therefore describe a more efficient procedure for finding Ψ and Φ, whose computational cost is only $O(k\log(k))$.

Let us examine two losses $\ell_{t,r}$ and $\ell_{t,s}$ such that $\ell_{t,r} \le \ell_{t,s}$ and there is no index j for which $\ell_{t,r} < \ell_{t,j} < \ell_{t,s}$. Then, all the sets $\Psi(\theta)$ for $\theta \in [\ell_{t,r}, \ell_{t,s})$ are identical, and equal $\{j \;:\; \ell_{t,j} \ge \ell_{t,r}\}$. Therefore, there are at most k different choices for $\Psi(\theta)$, which can be easily computed by sorting the losses. An analogous argument holds for the set Φ with respect to the values $\ell_{t,j} - C\|\mathbf{x}_{t,j}\|_2^2$. Furthermore, to enumerate all admissible sets $\Psi(\theta)$ and $\Phi(\theta)$ we need not examine their product space. Instead, let $\mathbf{q}$ denote the vector obtained by sorting the union of the sets $\{\ell_{t,j}\}_{j=1}^k$, $\{\ell_{t,j} - C\|\mathbf{x}_{t,j}\|_2^2\}_{j=1}^k$, and $\{0\}$ in ascending order. Extending the above rationale, the sets $\Psi(\theta)$ and $\Phi(\theta)$ are fixed for every $\theta \in [q_i, q_{i+1})$. We can examine every possible pair of candidates $\Psi(\theta), \Phi(\theta)$ by traversing the sorted vector $\mathbf{q}$ of critical values.

Concretely, define $\Psi(q_1) = \{1,\ldots,k\}$ and $\Phi(q_1) = \{1,\ldots,k\}$, and keep them sorted in memory. Use these sets to define θ and τ as described above, and check if the solution satisfies our constraints. If so, return this value of τ as the update step for the r-max loss. Otherwise, move on to the next value in $\mathbf{q}$ and evaluate the next pair of candidates. This procedure for choosing θ and τ implies that if more than one solution satisfies the constraints, we will choose the one encountered first, namely the one for which θ is the smallest. Indeed it can be verified that the smaller θ, the greater the value of the objective function in Equation (26). Given the sets $\Psi(q_i)$ and $\Phi(q_i)$, we can obtain the sets $\Psi(q_{i+1})$ and $\Phi(q_{i+1})$, and recalculate θ, by simply removing from $\Psi(q_i)$ every j for which

$\ell_{t,j} < q_{i+1}$ and removing from $\Phi(q_i)$ every j for which $\ell_{t,j} - C\|\mathbf{x}_{t,j}\|_2^2 < q_{i+1}$. This operation can be done efficiently since the sets $\Psi(q_i)$ and $\Phi(q_i)$ are sorted in memory.

8. Experiments with Text Classification

In this section, we demonstrate the effectiveness of the implicit multitask algorithm on large-scale text categorization problems. Throughout this paper, we have argued that when faced with multiple tasks in parallel, we can often do better than to learn each task individually. The goal of the first two experiments is to demonstrate that this is indeed the case. The third experiment demonstrates that the superiority of the implicit update algorithm, presented in Section 5, over the multitask Perceptron, presented in Sections 3 and 4.

We used the *Reuters Corpus Vol. 1*, which is a collection of over 800K news articles collected from the Reuters newswire over a period of 12 months, in 1996-1997. An average article contains approximately 240 words, and the entire corpus contains over half a million distinct tokens (not including numbers and dates). Each article is associated with *one or more* of 104 possible low-level categories.[1] On average, each article is associated with 1.5 low-level categories. The categorization problem induced by this corpus is referred to as a *multiclass-multilabel* (MCML) problem, since there are multiple possible classes (the 104 categories) and each article may assigned multiple labels. Examples of categories that appear in the corpus are: WEATHER, MONEY MARKETS, and UNEMPLOYMENT. The articles in the corpus are given in their original chronological order, and our goal is to predict the label, or labels, associated with each newly presented article. Our first experiment addresses this problem.

The Reuters corpus also defines 5 high-level meta-categories: CORPORATE/INDUSTRIAL, ECONOMICS, GOVERNMENT/SOCIAL, MARKETS, and OTHER. About 20% of the articles in the corpus are associated with more than one of the five meta-categories. After discarding this 20%, we are left with over 600K documents, each with a single high-level label. This induces a 5-class single-label classification problem. Our second experiment addresses this multiclass single-label problem.

We began by applying some mild preprocessing to the articles in the corpus, which included removal of punctuation, numbers, dates, and stop-words, and a global conversion of the entire corpus to lower-case. Then, each article was mapped to a real vector using a logarithmic bag-of-words representation. Namely, the length of each vector equals the number of distinct tokens in the corpus, and each coordinate represents one of these tokens. If a token appears s times in a given article, then the respective coordinate in the vector is set to $\log_2(1+s)$.

8.1 Multiclass Multilabel Categorization

We trained a separate binary classifier for each of the 104 low-level classes, using the implicit update algorithm presented in Section 5. Given an unseen article, each classifier predicts whether its respective category applies to that article or not. We ran our algorithm using both the L_1 norm and the L_∞ norm as the global loss function. In both cases, the user-defined parameter C was set to 10^{-3}.

The performance of the entire classifier ensemble on each article was evaluated in two ways. First, we examined whether the 104-classifier ensemble predicted the *entire* set of categories per-

[1] The original corpus specifies 126 labels which are organized in a hierarchical tree-structure. Of these labels, 104 are low-level categories, which correspond to leaves in the tree. The remaining labels are meta-categories which correspond to inner nodes in the tree.

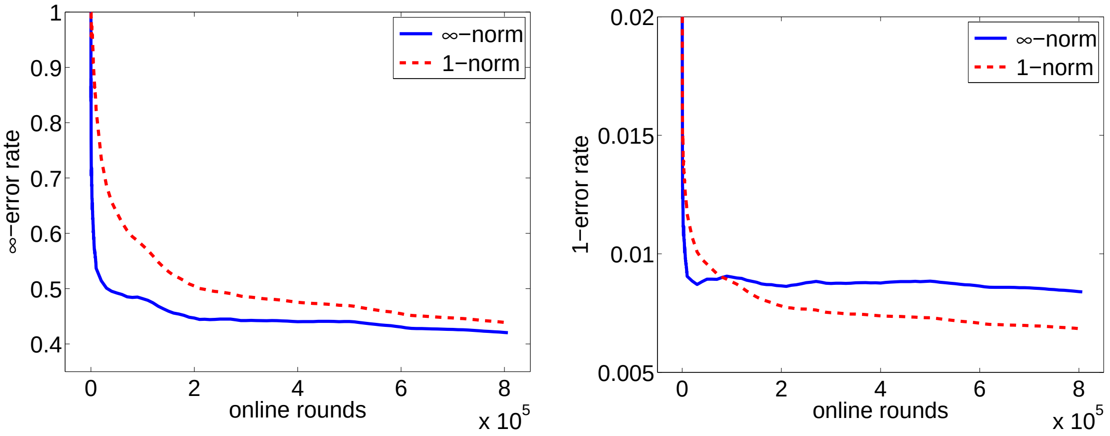

Figure 4: The ∞-error (left) and 1-error (right) error-rates attained by the implicit multitask algorithm using the L_∞ norm (solid) and the L_1 norm (dashed) global loss functions. Note that the two plots are on a very different scale: the two lines on the left-hand plot differ by approximately 3%, whereas the lines on the right-hand plot differ by approximately 0.05%.

fectly. An affirmative answer to this test implies that all 104 classifiers made correct predictions simultaneously. Formally, let $\mathbf{e}_t$ be the vector in $\{0,1\}^{104}$ such that $e_{t,j} = 1$ if and only if $y_{t,j}\mathbf{w}_{t,j} \cdot \mathbf{x}_{t,j} \leq 0$. In other words, $\mathbf{e}_t$ indicates which of the 104 binary classifiers made prediction mistakes on round t. Now define the *∞-error* suffered on round t as $\|\mathbf{e}_t\|_\infty$. Second, we assessed the fraction of categories for which incorrect binary predictions were made. Formally, define the *1-error* suffered on round t as $\|\mathbf{e}_t\|_1/104$. Both measures of error are reasonable, and one should be preferred over the other based on the specific requirements of the underlying application. Since each coordinate of ℓ_t upper-bounds the respective coordinate in $\mathbf{e}_t$, it holds that $\|\mathbf{e}_t\|_\infty \leq \|\ell_t\|_\infty$ and that $\|\mathbf{e}_t\|_1 \leq \|\ell_t\|_1$. Therefore, the L_∞ norm update seems to be a more appropriate choice for minimizing the ∞-error, while the L_1 norm update is the more appropriate choice for minimizing the 1-error. Our experiments confirm this intuitive argument.

The results of our experiments are summarized in Figure 4. The left-hand plot in the figure shows the ∞-error-rate of the L_∞ norm and L_1 norm multitask updates, as the number of examples grows from zero to 800K. The figure clearly shows that the L_∞ norm algorithm does a better job throughout the entire online learning process. The advantage of the L_∞ norm algorithm diminishes as more examples are observed.

The right-hand plot in Figure 4 compares the 1-error-rate of the two updates. In this case, the L_∞ norm update initially takes the lead, but is quickly surpassed by the L_1 norm update. The fact that the L_1 norm update ultimately gains the advantage coincides with our initial intuition. The reason why the L_∞ norm update outperforms the L_1 norm update at first can also be easily explained. The L_1 norm update is quite aggressive, as it modifies every binary classifier that suffers a positive individual loss on every round. Moreover, the L_1 norm update enforces the constraint $\|\tau_t\|_\infty \leq C$. On the other hand, the L_∞ norm update is more cautious, since it enforces the stricter constraint $\|\tau_t\|_1 \leq C$. The aggression of the L_1 norm update causes its initial behavior to be somewhat erratic.

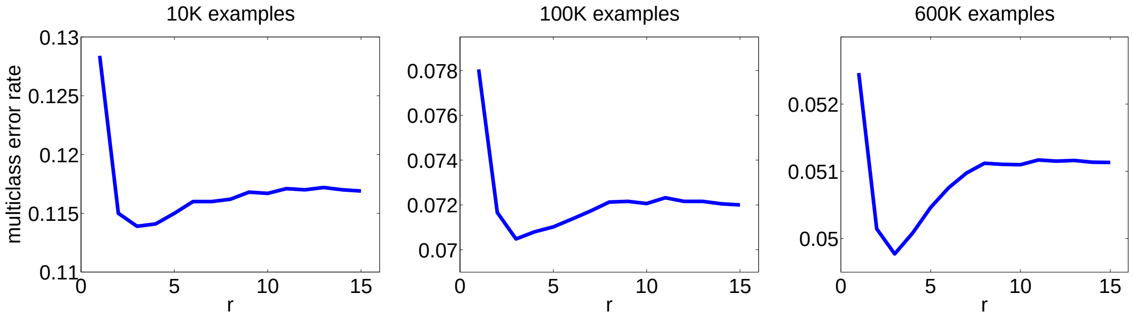

Figure 5: The multiclass error rate of the online ECOC-based classifier, using a 15 column code matrix, with various r-max norms, after observing 10K, 100K, and 600K examples.

At first, many of the L_1 norm updates actually move the classifier ensemble away from its target. Inevitably, it takes the L_1 norm classifier slightly longer to find its path.

8.2 Multiclass Meta-Categorization with ECOC and r-max Norms

Following one of the motivating examples given in the introduction, we used the ECOC method (Dietterich and Bakiri, 1995) to reduce the 5 high-level meta-categories classification task from the Reuters corpus to multiple binary classification tasks. We used the 5×15 Hadamard code matrix, defined as follows:

$$M = \begin{pmatrix} + & + & + & + & + & + & + & + & + & + & + & + & + & + & + \\ + & + & + & + & + & + & + & - & - & - & - & - & - & - & - \\ + & + & + & - & - & - & - & + & + & + & + & - & - & - & - \\ + & - & - & + & + & - & - & + & + & - & - & + & + & - & - \\ - & + & - & + & - & + & - & + & - & + & - & + & - & + & - \end{pmatrix} .$$

This code matrix is derived by taking all 2^4 possible 5-coordinate columns with $+$ in the first position, except for the all-plus column. This is the largest 5-row code matrix that does not induce redundant or trivial binary classification problems. The distance between any two rows of the matrix is 8, therefore this code is guaranteed to correct 4 binary prediction mistakes. We can determine if more than 4 binary mistakes are made on round t by comparing the fifth largest element of ℓ_t with 1. As mentioned in the introduction, taking the fifth largest loss does not constitute a norm, and cannot be used as a global loss within our setting. However, a norm with a similar flavor is the r-max norm, with $r = 5$. Our experiments show that it is actually advantageous to be slightly over-cautious, by setting r to 3 or 4.

The results of our experiments are summarized in Figure 5. We trained 15 binary classifiers, one per each column of M, using the implicit update algorithm presented in Section 5. We used the r-max norm as the algorithm's global loss function, with r set to every integer value between 1 and 15. For each example, all 15 binary classifiers made predictions, and M was used to decode a multiclass prediction, as described in Dietterich and Bakiri (1995). A multiclass error occurs if the predicted label differs from the true label. In Figure 5 we depict the average number of errors that occurred after observing 10K, 100K, and 600K examples, for each value of r. We can see that

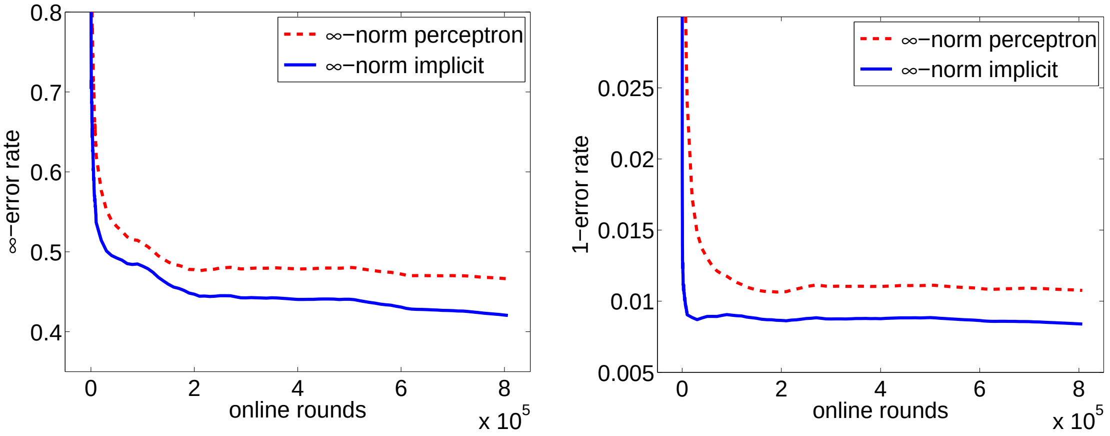

Figure 6: The ∞-error (left) and 1-error (right) attained by the multitask Perceptron (dashed) and the implicit update algorithm (solid) when using the L_∞ norm as a global loss function.

using either the L_1 norm ($r = 15$) or the L_∞ norm ($r = 1$) is suboptimal, and the best performance is consistently reached by setting r to be slightly smaller than half the code distance. Although the theoretically motivated choice of $r = 5$ is not the best, it still yields better results than the two extreme choices, $r = 1$ and $r = 15$.

When we replaced the Hadamard code matrix with the One-vs-Rest code matrix, defined by $2I - 1$ (where I is the 5×5 identity matrix and 1 is the 5×5 all-ones matrix) then the multiclass error after observing 600K examples increases from 5% to around 8%. This justifies using the ECOC method in the first place.

We conclude this experiment by noting that although setting $r = 1$ produces the largest number of multiclass prediction mistakes, it still delivers the best performance if we evaluate the 15 classifier ensemble using the ∞-error defined above.

8.3 The Implicit Update vs. the Multitask Perceptron

From a loss minimization standpoint, Theorem 9 proves that the implicit update, presented in Section 5, is at least as good as the multitask Perceptron variants, presented in Secs. 3 and 4. The following experiment demonstrates that the implicit update is also superior in practice.

We repeated the multitask multi-label experiment described in Section 8.1, using the multitask Perceptron in place of the implicit update algorithm. The infinite horizon extension discussed in Section 4 does not have a significant effect on empirical performance, so we consider only the finite horizon version of the multitask Perceptron, described in Section 3.

When the global loss function is defined using the L_1 norm, both the implicit update and the multitask Perceptron update decouple to independent updates for each individual task. In this case, both algorithms are very similar, their empirical performance is almost identical, and the comparison between them is not very interesting. Therefore, we focus on a global loss defined by the L_∞ norm.

A comparison between the performance of the implicit update and the multitask Perceptron update, both using the L_∞-norm loss, is given in Figure 6. The plot on the left-hand side of the figure compares the two algorithms' ∞-error-rate, and the plot on the right-hand side of the figure compares their 1-error-rate. The implicit algorithm holds a clear lead over the multitask Perceptron with respect to both error measures, throughout the learning process. These results give empirical validation to the formal comparison of the two algorithms.

9. Discussion

When faced with several online tasks in parallel, it is not always best to distribute the learning effort evenly. In many cases, it may be beneficial to allocate more effort to tasks when they are seen to play “key” roles. In this paper, we presented an online algorithmic framework that does precisely that. The priority given to each task is governed by its relative performance and by the choice of a global loss function.

We presented three families of algorithms, each of which includes an algorithm for every global loss defined by an absolute norm. The first two families are illustrative and theoretically appealing. The third family of algorithms uses the most sophisticated update of the three, and is the one recommended for practical use. We demonstrated the superior performance of the third family of algorithms empirically.

We showed that, in the worst case, the finite horizon multitask Perceptron of Section 3 and the implicit update algorithm of Section 5 both perform asymptotically as well as the best fixed hypothesis ensemble. In other words, these algorithms are no-regret algorithms with respect to any global loss function defined by an absolute norm. The same cannot be said for the naive alternative, where we use multiple independent single-task learning algorithms to solve the multitask problem. We also demonstrated the benefit of the multitask approach over the naive alternative on two large-scale text categorization problems.

Throughout the paper, we assumed that the multiple online tasks are perfectly synchronized, and that a complete k-tuple of examples is observed on every round. This is indeed the case in each of the concrete examples described in the introduction and empirically tested in our experiments. However, in other real-world situations, this may not be the case. Namely, there could occur situations where not all of the tasks are active on every single round. In other words, there may be a subset of “dormant tasks” on each round. For example, say that we are operating an online store and that we have multiple registered customers. Each product in our store is represented by a feature vector, and we train an individual binary classifier for each of our customers. When costumer j visits a product-page on our website, the respective classifier is used to predict whether that customer intends to purchase the product or not. The prediction is then used to decide whether or not to lure the customer away from that page. This setting induces a natural online multitask learning problem. Moreover, only a fraction of the customers is online at any given moment. We consider the tasks of those customers that are not online to be dormant or inactive tasks. At a first glance, the inactive tasks setting may seem to be more complicated than the fully synchronized setting discussed throughout the paper. However, our algorithms and analysis accommodate this extension quite naturally. We simply need to define $\ell_{t,j} = 0$ for every inactive task and apply the multitask update verbatim. Due to the conservativeness assumption, the hypotheses of the inactive tasks will be left intact. Additionally, note that all of the norms discussed in this paper have the property that $\|\mathbf{v}\| = \|\mathbf{v}'\|$, where $\mathbf{v}'$ is the vector obtained by removing all of the zero entries from $\mathbf{v}$. Therefore, we can imagine that the length of the vector ℓ_t changes from round to round, and that the update on each round is applied as if the tasks that are sleeping on that round never existed in the first place. We would also like to note that, although our presentation focuses on multiple binary classification tasks, our algorithms and techniques can be adapted to other online learning problems as well. Specifically, a multitask implicit update can be derived for regression and uniclass problems using ideas from Crammer et al. (2006).

The next-step would be to extend our framework from absolute norms to general norms. For example, the family of Mahalanobis norms, defined by $\|\mathbf{z}\|^2 = \mathbf{z}^\top P\mathbf{z}$ (where P is a positive definite matrix) includes norms that are not absolute but which could have interesting applications in our setting. More generally, there exist meaningful global loss functions which are not norms at all.

Another interesting research direction would be to return to the roots of statistical multitask learning, and to try to model generative similarities between the multiple tasks within the online framework. In our work, we completely disregarded any relatedness between the multiple tasks, and only considered the shared consequences of errors. In the game-theoretic spirit of online learning, modeling these similarities would have to be done without making statistical assumptions on the data source.

Appendix A. The K-Method of Norm Interpolation

In this section, we briefly survey Peetre's K-method of norm interpolation. This method takes a pair of norms and smoothly interpolates between them, producing a new family of norms which can be used in our setting. An example of such an interpolation is the family of r-max norms, previously mentioned in this paper. The main practical purpose of this section is to prove that the dual of the r-max norm takes the form given in Equation (3). We do not present the K-method in all its generality, but rather focus only on topics which are relevant to the online multitask learning setting. The interested reader is referred to Bennett and Sharpley (1998) for a more detailed account of interpolation theory.

We begin by presenting Peetre's K-functional and J-functional, and proving that they induce dual norms. Let $\|\cdot\|_{p_1} : \mathbb{R}^k \rightarrow \mathbb{R}_+$ and $\|\cdot\|_{p_2} : \mathbb{R}^k \rightarrow \mathbb{R}_+$ be two p-norms, and let $\|\cdot\|_{q_1}$ and $\|\cdot\|_{q_2}$ be their respective duals. The K-functional with respect to p_1 and p_2, and with respect to the constant $\alpha > 0$, is defined as

$$\|\mathbf{v}\|_{K(p_1,p_2,\alpha)} \;=\; \min_{\mathbf{w}+\mathbf{z}=\mathbf{v}} \Big(\|\mathbf{w}\|_{p_1} + \alpha\|\mathbf{z}\|_{p_2}\Big) \quad .$$

The J-functional with respect to q_1, q_2, and with respect to the constant $\beta > 0$, is defined as

$$\|\mathbf{u}\|_{J(q_1,q_2,\beta)} \;=\; \max\Big\{\|\mathbf{u}\|_{q_1},\; \beta\,\|\mathbf{u}\|_{q_2}\Big\} \quad .$$

The J-functional is obviously a norm: positivity and linearity follow immediately from the fact that $\|\cdot\|_{q_1}$ and $\|\cdot\|_{q_2}$ posses these properties. The triangle inequality follows from

$$\begin{aligned}
\|\mathbf{v}+\mathbf{u}\|_{J(q_1,q_2,\beta)} &= \max\Big\{\|\mathbf{v}+\mathbf{u}\|_{q_1},\; \beta\,\|\mathbf{v}+\mathbf{u}\|_{q_2}\Big\} \\
&\leq \max\Big\{\|\mathbf{v}\|_{q_1} + \|\mathbf{u}\|_{q_1},\; \beta\|\mathbf{v}\|_{q_2} + \beta\|\mathbf{u}\|_{q_2}\Big\} \\
&\leq \max\Big\{\|\mathbf{v}\|_{q_1},\; \beta\|\mathbf{v}\|_{q_2}\Big\} \;+\; \max\Big\{\|\mathbf{u}\|_{q_1},\; \beta\|\mathbf{u}\|_{q_2}\Big\} \\
&= \|\mathbf{v}\|_{J(q_1,q_2,\beta)} \;+\; \|\mathbf{u}\|_{J(q_1,q_2,\beta)} \quad .
\end{aligned}$$

Since the J-functional is defined with respect to two absolute norms, it too is an absolute norm.

Instead of explicitly proving that $\|\cdot\|_{K(p_1,p_2,\alpha)}$ is also a norm, we prove that it is the dual of $\|\cdot\|_{J(q_1,q_2,\beta)}$ when $\alpha = 1/\beta$. Since the dual of an absolute norm is itself an absolute norm, and since the dual of the dual norm is the original norm (Horn and Johnson, 1985), our proof implies that $\|\cdot\|_{K(p_1,p_2,\alpha)}$ is indeed a norm, that it is absolute, and that its dual is $\|\cdot\|_{J(q_1,q_2,1/\alpha)}$.

Theorem 10 *Using the notation defined above,*

$$\|\cdot\|^*_{J(q_1,q_2,\beta)} \equiv \|\cdot\|_{K(p_1,p_2,1/\beta)} \quad .$$

Proof We abbreviate $\|\mathbf{v}\|_J = \|\mathbf{v}\|_{J(q_1,q_2,\beta)}$ and $\|\mathbf{v}\|_K = \|\mathbf{v}\|_{K(p_1,p_2,1/\beta)}$ throughout the proof. First, we show that $\|\mathbf{v}\|^*_J \le \|\mathbf{v}\|_K$ for all $\mathbf{v} \in \mathbb{R}^k$. Let $\mathbf{v},\mathbf{w}$ and $\mathbf{z}$ be vectors in $\mathbb{R}^k$ such that $\mathbf{v} = \mathbf{w}+\mathbf{z}$. Then for any $\mathbf{u} \in \mathbb{R}^k$, we can use Hölder's inequality to obtain

$$\begin{aligned} \mathbf{u}\cdot\mathbf{v} &= \mathbf{u}\cdot\mathbf{w} + \mathbf{u}\cdot\mathbf{z} \\ &\le \|\mathbf{u}\|_{q_1}\|\mathbf{w}\|_{p_1} + \|\mathbf{u}\|_{q_2}\|\mathbf{z}\|_{p_2} \quad . \end{aligned}$$

By definition, it holds that

$$\|\mathbf{u}\|_{q_1} \le \|\mathbf{u}\|_J \quad \text{and} \quad \|\mathbf{u}\|_{q_2} \le \frac{1}{\beta}\|\mathbf{u}\|_J \quad ,$$

and so

$$\mathbf{u}\cdot\mathbf{v} \le \left(\|\mathbf{w}\|_{p_1} + \frac{1}{\beta}\|\mathbf{z}\|_{p_2}\right)\|\mathbf{u}\|_J \quad .$$

Since the only restriction on $\mathbf{u},\mathbf{v},\mathbf{w}$ and $\mathbf{z}$ is that $\mathbf{v} = \mathbf{w}+\mathbf{z}$, we can fix $\mathbf{v}$, choose $\mathbf{u}$ to be the vector which maximizes the left-hand side above subject to $\|\mathbf{u}\|_J \le 1$, and choose $\mathbf{w}$ and $\mathbf{z}$ which minimize the right-hand side above subject to $\mathbf{v} = \mathbf{w}+\mathbf{z}$. This results in

$$\max_{\mathbf{u}\in\mathbb{R}^k:\|\mathbf{u}\|_J\le 1} \mathbf{u}\cdot\mathbf{v} \le \min_{\mathbf{w}+\mathbf{z}=\mathbf{v}} \left(\|\mathbf{w}\|_{p_1} + \frac{1}{\beta}\|\mathbf{z}\|_{p_2}\right) \quad .$$

The left-hand side above is the formal definition of $\|\mathbf{v}\|^*_J$, the right-hand side is the definition of $\|\mathbf{v}\|_K$, and we have proven that $\|\mathbf{v}\|^*_J \le \|\mathbf{v}\|_K$.

To prove the opposite direction, fix $\mathbf{v}$ and let $\mathbf{u}$ be the vector with $\|\mathbf{u}\|_J \le 1$ which maximizes $\mathbf{u}\cdot\mathbf{v}$. We now consider two cases. If $\|\mathbf{u}\|_{q_1} \ge \beta\|\mathbf{u}\|_{q_2}$ then

$$\|\mathbf{v}\|^*_J = \max_{\mathbf{u}:\|\mathbf{u}\|_{q_1}\le 1} \mathbf{u}\cdot\mathbf{v} \quad .$$

Using the duality of $\|\cdot\|_{q_1}$ and $\|\cdot\|_{p_1}$, the right hand-side above equals $\|\mathbf{v}\|_{p_1}$. Since we can choose $\mathbf{w} = \mathbf{v}$ and $\mathbf{z} = 0$, it certainly holds that

$$\|\mathbf{v}\|_{p_1} \ge \min_{\mathbf{w}+\mathbf{z}=\mathbf{v}} \left(\|\mathbf{w}\|_{p_1} + \frac{1}{\beta}\|\mathbf{z}\|_{p_2}\right) = \|\mathbf{v}\|_K \quad .$$

On the other hand, if $\|\mathbf{u}\|_{q_1} \le \beta\|\mathbf{u}\|_{q_2}$ then

$$\|\mathbf{v}\|^*_J = \frac{1}{\beta}\max_{\mathbf{u}:\|\mathbf{u}\|_{p_2}\le 1} \mathbf{u}\cdot\mathbf{v} \quad .$$

Using the duality of $\|\cdot\|_{q_2}$ and $\|\cdot\|_{p_2}$, the right hand-side above equals $\frac{1}{\beta}\|\mathbf{v}\|_{p_2}$. Since we can choose $\mathbf{w}=0$ and $\mathbf{z}=\mathbf{v}$, it holds that

$$\frac{1}{\beta}\|\mathbf{v}\|_{p_2} \;\geq\; \min_{\mathbf{w}+\mathbf{z}=\mathbf{v}} \left(\|\mathbf{w}\|_{p_2}+\frac{1}{\beta}\|\mathbf{z}\|_{p_2}\right) \;=\; \|\mathbf{v}\|_K \quad .$$

Overall, we have shown that $\|\mathbf{v}\|_J^* \geq \|\mathbf{v}\|_K$. ■

The r-max norm discussed in the paper is an instance of the K-functional, and can be defined as

$$\|\mathbf{v}\|_{r\text{-max}} \;=\; \|\mathbf{v}\|_{K(1,\infty,r)} \quad .$$

To see why this is true, let ϕ be the absolute value of the r'th absolutely largest coordinate in $\mathbf{v}$. Now define for each $1 \leq j \leq k$

$$w_j = \operatorname{sign}(v_j)\max\{0, |v_j|-\phi\} \quad \text{and} \quad z_j \;=\; \operatorname{sign}(v_j)\min\{|v_j|, \phi\} \quad .$$

Note that $\mathbf{w}+\mathbf{z}=\mathbf{v}$, and that

$$\|\mathbf{v}\|_{r\text{-max}} \;=\; \|\mathbf{w}\|_1 + r\|\mathbf{z}\|_\infty \quad .$$

This proves that $\|\mathbf{v}\|_{r\text{-max}} \geq \|\mathbf{v}\|_{K(1,\infty,r)}$.

Turning to the opposite inequality, let $\pi(1),\ldots,\pi(r)$ be the indices of the r absolutely largest elements of $\mathbf{v}$, and let $\mathbf{w}$ and $\mathbf{z}$ be vectors such that $\mathbf{w}+\mathbf{z}=\mathbf{v}$. We now have that

$$\begin{aligned}
\|\mathbf{v}\|_{r\text{-max}} &= \sum_{j=1}^{r} |v_{\pi(j)}| \\
&= \sum_{j=1}^{r} |w_{\pi(j)} + z_{\pi(j)}| \\
&\leq \sum_{j=1}^{r} |w_{\pi(j)}| \;+\; \sum_{j=1}^{r} |z_{\pi(j)}| \\
&\leq \sum_{j=1}^{r} |w_{\pi(j)}| \;+\; r \max_{j=1,\ldots,r} |z_{\pi(j)}| \\
&\leq \sum_{j=1}^{k} |w_j| \;+\; r \max_{j=1,\ldots,k} |z_j| \;=\; \|\mathbf{w}\|_1 \;+\; r\|\mathbf{z}\|_\infty \quad .
\end{aligned}$$

The above holds for any $\mathbf{w}$ and $\mathbf{z}$ which sum to $\mathbf{v}$, and specifically to those which minimize $\|\mathbf{w}\|_1 + r\|\mathbf{z}\|_\infty$. We conclude that $\|\mathbf{v}\|_{r\text{-max}} \leq \|\mathbf{v}\|_{K(1,\infty,r)}$, and therefore $\|\mathbf{v}\|_{r\text{-max}} = \|\mathbf{v}\|_{K(1,\infty,r)}$.

Finally, we calculate an upper bound on the remoteness of $\|\cdot\|_{J(q_1,q_2,\beta)}$. This enables us to obtain concrete loss bounds for interpolation norms from the theorems proven in this paper. Recall that

$$\rho(\|\cdot\|_{J(q_1,q_2,\beta)}, k) \;=\; \max_{\mathbf{u}\in\mathbb{R}^k} \frac{\|\mathbf{u}\|_2}{\|\mathbf{u}\|_{J(q_1,q_2,\beta)}} \quad .$$

Using the definition of the J-functional, the above becomes

$$\max_{\mathbf{u}\in\mathbb{R}^k} \min\left\{\frac{\|\mathbf{u}\|_2}{\|\mathbf{u}\|_{q_1}}, \frac{\|\mathbf{u}\|_2}{\beta\|\mathbf{u}\|_{q_2}}\right\} \quad .$$

Using the weak minimax theorem, we can upper-bound the above by

$$\min\left\{\max_{\mathbf{u}\in\mathbb{R}^k}\frac{\|\mathbf{u}\|_2}{\|\mathbf{u}\|_{q_1}},\max_{\mathbf{u}\in\mathbb{R}^k}\frac{\|\mathbf{u}\|_2}{\beta\|\mathbf{u}\|_{q_2}}\right\} \quad .$$

Once again using the definition of remoteness, the above can be rewritten as

$$\min\left\{\rho(\|\cdot\|_{q_1},k),\ \frac{\rho(\|\cdot\|_{q_2},k)}{\beta}\right\} \quad .$$

Using Lemma 2, we can obtain an explicit upper bound on the remoteness of any interpolation of p-norms.

References

J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12: 149–198, 2000.

S. Ben-David and R. Schuller. Exploiting task relatedness for multiple task learning. In *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory*, 2003.

C. Bennett and R. Sharpley. *Interpolation of Operators*. Academic Press, 1998.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

O. Chapelle and Z. Harchaoui. A machine learning approach to conjoint analysis. In *Advances in Neural Information Processing Systems*, volume 17, 2005.

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

K. Crammer and Y. Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991, 2003.

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, Mar 2006.

T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, January 1995.

T. Evgeniou, C.Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

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

R. Herbrich, T. Graepel, and K. Obermayer. Large marging rank boundaries for ordinal regression. In A. Smola, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 2000.

T. Heskes. Solving a huge number of silmilar tasks: A combination of multitask learning and a hierarchical bayesian approach. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 233–241, 1998.

R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

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

A. B. J. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume XII, pages 615–622, 1962.

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

I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the Twenty-First International Conference on Machine Learning*, 2004.