Matrix regularization techniques for online multitask learning

Alekh Agarwal
Alexander Rakhlin
Peter Bartlett

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2008-138
http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-138.html

October 23, 2008

Matrix regularization techniques for online multitask learning

Alekh Agarwal*
Computer Science Division
UC Berkeley
alekh@cs.berkeley.edu

Alexander Rakhlin
Computer Science Division
UC Berkeley
rakhlin@cs.berkeley.edu

Peter L. Bartlett
Computer Science Division
Department of Statistics
UC Berkeley
bartlett@cs.berkeley.edu

Abstract

In this paper we examine the problem of prediction with expert advice in a setup where the learner is presented with a sequence of examples coming from different tasks. In order for the learner to be able to benefit from performing multiple tasks simultaneously, we make assumptions of task relatedness by constraining the comparator to use a lesser number of *best* experts than the number of tasks. We show how this corresponds naturally to learning under spectral or structural matrix constraints, and propose regularization techniques to enforce the constraints. The regularization techniques proposed here are interesting in their own right and multitask learning is just one application for the ideas. A theoretical analysis of one such regularizer is performed, and a regret bound that shows benefits of this setup is reported.

1 Introduction

The problem of multitask learning is a scenario where the learner receives examples drawn from more than one task. As algorithms for single-task problems are readily available, the simplest approach is to solve each of the tasks independently of the others. However, if tasks are *related*, ignoring the common structure means throwing out useful information. From the algorithmic point of view, the hallmark of multitask learning is developing ways of exploiting task relatedness. From the theoretical point of view, the goal is to quantify the improvement, taking as a baseline performance when tasks are learned separately. In this paper, we provide a new algorithm and quantify the gain of learning the tasks together. We note that the algorithm we present is interesting in its own right and can be employed in settings beyond multitask learning.

The multitask learning problem has seen a lot of work in recent years (e.g. [4], [3], [2]). A common theme of all these approaches is to model the notion of task relatedness via an assumption about the low rank of the data matrix. However, these approaches often result in non-convex optimization problems, which cannot be solved exactly in a computationally efficient manner. Also, no significant theoretical analysis has been done in this setup.

In the online setup, this problem was looked at in [7], [1] and recently in [5]. In online multitask learning problem with experts, the learner receives a task id and a vector of the losses of each expert at every time step. The loss of the learner is the expected loss under his distribution over the experts for that task. The notion of task relatedness is a little different in this setup, and will be described at length in the following section.

*Corresponding author.

The authors in [1] present an experts algorithm similar to weighted majority for obtaining the optimal regret bound in this problem. This optimal algorithm however, involves an NP-Hard computation, and is approximated by an MCMC procedure. Cavallanti et al [5] used matrix regularization approaches that were used in the batch setup to obtain efficient algorithms, but their regret bound scales much worse than the optimal, in the worst case incurring a linear dependence on the number of tasks.

In this paper, we will look at some new matrix regularization ideas that are more suited to this learning problem. In particular, we will describe spectral and structural matrix regularization approaches, and show how the latter allows us to obtain efficient low regret algorithms.

Summary of results: In this paper we obtain a computationally efficient algorithm whose regret scales no worse than $O(\sqrt{KmT\log N})$ for learning K tasks with N experts for each, with parameter m quantifying task similarty. We will show that in some interesting regimes of K, N values, this comes quite close to the optimal regret. This is a significant improvement over the result of Cavallanti et al [5] whose algorithm yields an upper bound on the regret scaling as $O((K-m)\sqrt{T\log\min\{K,N\}})$. The latter bound can be linear in the number of tasks in the worst case, implying little gain over the baseline of learning tasks independently.

2 Setup

In this section, we introduce some notation to be used throughout the paper and provide a precise statement of the problem.

We denote vectors by lower case letters such as u, w, x, l. Upper case letters typically denote matrices. We will usually use U, W to refer to matrices used by the comparator and player respectively, and these matrices will be non-negative with rows adding up to 1 unless otherwise specified. W^i will be used to denote the i_{th} column of matrix W. The notations $\mathcal{S}_m$ and $\mathcal{S}_{\leq m}$ introduced below will be used interchangibly for the comparator class specified in terms of experts or the comparator matrix U. Pairs p, q and r, s will be used to refer to conjugate exponents, i.e. $1/p + 1/q = 1$ and $1/r + 1/s = 1$.

2.1 Problem Specification

Let us now formalize the game by describing the notions of tasks and relatedness. Since the only assumption that relates the sequence in a classical online setup is the restriction of comparator to a *single* best expert, it is natural to define a task as a set of examples that use the same best comparator. Of course, this assumption is meaningless unless the learner knows which examples would be using the same comparator. Hence, we assume that the learner, along with the predictions of the experts, is also told the identity of the task from which these predictions were generated. In this article, we define a setup where there are N different experts whose predictions we receive at each step, for one of K different tasks.

For the case of a single task, the minimax optimal regret bound is $\sqrt{T/2\ln N}$. Without any further assumption on tasks, it is clear that $K\sqrt{T/2\ln N}$ is the best regret that can be achieved. However, we hope to do better when the tasks are related. One way to formalize this is by saying that the *actual* number of different tasks is just $m \ll K$. This amounts to saying that the comparator is allowed to have just m instead of K different predictors. However, the learner still gets a task identity between $1, \ldots, K$. The aim of the learner is to discover the task similarity so as to get a regret bound with some dependence on m. To get an idea of the optimal dependence on m and other parameters, we need to consider the effective comparator class of our problem.

2.2 Comparator class

In the expert setting, the regret is typically defined as:

$$\sum_{t=1}^{T} \hat{L}^t - \min_i \sum_{t=1}^{T} L_i^t \tag{1}$$

where $\hat{L}^t$ is the loss of the learner at time t. In the multitask problem, we wish to generalize this to m different comparators. Let $[K] = \{1, \ldots, K\}$ and $\mathcal{S}_m := \{S \subset [K] : |S| = m\}$. Then clearly, any comparator from the class $\mathcal{S}_m$ uses only m different experts, and thus forms our comparator class. The regret of our problem is defined as:

$$R_T = \sum_{t=1}^{T} \hat{L}^t - \min_{S \in \mathcal{S}_m} \sum_{k=1}^{K} \min_{i \in S} \sum_{t \in T_k} L_{i,k}^t \tag{2}$$

where $T_k = \{t : k_t = k\}$ and k_t is the task id at time t. In the single task setting, the regret bound contains the factor $\ln N$, which is seen to be the log cardinality of the comparator class for that problem (as the optimal comparator picks just one of the experts). Then it is natural to ask if we can compute the cardinality of this comparator class. The hope would be to obtain regret bounds scaling with the log of its size again.

It is not too hard to see that $\mathcal{S}_m \leq \binom{N}{m} m^K$. In fact, it is shown in [1] that this estimate is asymptotically of the right order, and thus $\log |\mathcal{S}_m| = \Theta\left(m \log \frac{N}{m} + K \log m\right)$. The authors further showed that the weighted majority algorithm can be easily adapted to instead learn distributions over the elements of $\mathcal{S}_m$ to indeed attain a regret based on this quantity. However, the algorithm needs to maintain distributions over an exponentially large class of experts now, and it was shown that computing and updating these weights is NP-Hard.

We can now setup the game of online multitask learning as in Figure 1

1: **for** $t = 1$ to T **do**
2: Aversary gives task id k_t.
3: Player specifies its distribution over experts for the task $\hat{p}_t$.
4: Pleayer incurs the loss $\hat{p}_t \ell_t$ for the loss vector ℓ_t generated by the adversary.
5: **end for**

Figure 1: The online multitask learning game

A slight generalization to the above setup will be used in the later sections of this paper. Consider the extended comparator class $\mathcal{S}_{\leq m} := \{S \subset [K] : |S| \leq m\}$. By a similar argument as above, the size of this class bounded by $N^m m^K$. Thus $\log |\mathcal{S}_{\leq m}| = \Theta\left(m \log N + K \log m\right)$.

3 Matrix Regularization for multitask learning

The main idea behind applying matrix regularization techniques to the multitask learning problem starts with first representing the comparators, predictors and examples as $K \times N$ matrices instead of vectors. As we try to specify a distribution over experts for each task, all the entries of the matrix have to be non-negative, and each row should add up to 1. The matrix for the comparator can be seen as a 0-1 matrix with only one non-zero entry per row which identifies the optimal expert for each task. Clearly at most m columns in this matrix can have non-zero entries by the assumptions of the previous section, if the comparator is in the class $\mathcal{S}_{\leq m}$. This is because every non-zero column of this matrix represents an expert used for at least one task. So to make sure that the total number of experts used across all tasks is small, we have to keep the number of non-zero columns small. The learner's matrix represents the K vectors corresponding to its predictive distributions over the experts for the various tasks. Finally, for homogeneity, we represent the vector of losses at each time as the $K \times N$ matrix X_t, where X_t has only one non-zero row corresponding to the task from which the example is drawn.

Now consider the comparator matrix. This matrix has at most m non-zero columns. Assuming $m < N, K$, the matrix, thus, has a rank of at most m. When competing against a comparator of low rank, it makes sense for the learner also to restrict itself to only those sets of distributions over tasks which mostly give rise to low rank matrices. This is intuitively reasonable, as this allows the learner to restrict itself to a smaller subclass of matrices, and thus exploit the particular structure of the comparator. As the regret of online algorithms

typically scales with the size of the class that they search over, there is hope that such an algorithm looking over a small class will also achieve a lower regret guarantee.

In fact, if the learner maintains a prediction matrix W of rank m, then the matrix can be factorized as $W = AB$ where A is $K \times m$ and B is $m \times N$. The matrix A can be seen as specifying a mapping from tasks to their *true* task ids (of which there are only m), while B gives a map from these to the best expert for each *true* task. However, it is immediate that the low rank assumption allows more general structures. In particular, a task could be a linear combination of two other tasks, and this would still keep the matrix low rank. This generality is also quite intuitive from a point of view of a notion of task-relatedness.

Since, we are doing online updates, if we have an update such that once initialized with a rank m matrix, it will always keep the matrix rank smaller than or equal to m, then we could exploit such an update in this problem by initializing the learner at a rank m matrix. This idea has been exploited for learning low rank kernel matrices in an online fashion in [8]. The key idea is to define suitable Bregman divergences over the space of matrices. For two square symmetric positive semidefinite (psd) matrices X, Y define the von Neumann and Burg divergences, respectively, as

$$D_{VN}(X, Y) = Tr(X \log X - X \log Y - X + Y) \tag{3}$$

$$D_{Burg}(X, Y) = Tr(XY^{-1}) - \text{logdet}(XY^{-1}) - n \tag{4}$$

Note that the Burg divergence is a natural generalization of log-barrier to square symmetric matrices, whereas the von Neumann divergence is the analogue of the entropy function.

The interesting fact is that if the matrix Y has a rank m, then any matrix X with a finite Burg divergence (D_{Burg}) from Y has a rank m as well. This is seen by writing the divergence as a function of eigenvalues. For a finite von Neumann divergence (D_{VN}) X needs to have a rank $\leq m$. This means that if the online updates are done by minimizing the loss plus one of these divergences, then the rank constraints are automatically enforced.

Unfortunately, these divergences are only defined for square symmetric matrices. An obvious way to extend this to rectangular matrices is by applying these divergences to $WW^\top$ or $(WW^\top)^{1/2}$ which are both square symmetric and psd. Unfortunately, this leads to an algorithm for which updates are hard to analyze. In this article, we will not explore this thought further, although it is conceivable that this approach might enjoy the a near optimal regret bound.

A remark about the structure of the comparator set is in order at this point. Note that the rank constraint is a non-convex constraint; the rank of a sum of two matrices can be as large as the sum of their ranks. In fact, it is not hard to construct full rank matrices using matrices from $\mathcal{S}_{\leq m}$. So if we naively try to maintain a low rank by restricting our online optimizations to this set, we could be in trouble. While the divergences given above provide a way of optimization under rank constraints in the case of square matrices, we cannot in general extend any algorithm to work within this subset by forcing it to optimize over just $\mathcal{S}_{\leq m}$. A natural strategy that is often used in such problems is to instead augment the objective function with a regularizer which takes small values on the regions of choice– matrices in $\mathcal{S}_{\leq m}$ in this case. While performing regularized optimization with such functions doesn't guarantee that the player will stay in the set, it forces a preference for staying in the set. Below we will see some matrix norms that try to fulfil this intuition. While it is possible to get to arbitrary distributions over experts by performing optimization under these regularizers, we will see that the preference they model to stay in the set is strong enough and suffices to obtain non-trivial regret guarantees.

3.1 Structural and spectral matrix norms

In the previous section, while describing the constraints we want to impose on the learner and comparator, we went back and forth between the idea that the matrix describing these distributions can be rank m or can have at most m non-zero columns. While the former argument leads to the idea of regularizing the eigenvalues so that a small number of them are non-zero, the latter encourages a more direct regularization of the entries of the matrix in such a way that the number of non-zero columns in the matrix is small.

We refer to norms (or general regularizing functions) acting on the eigenvalues as *spectral* norms, as they only depend on the spectrum of the matrix. The norms (regularizers) that act directly on the entries of the matrix to enforce a specific structure, such as a small number of non-zero columns, are referred to as *structural*. While spectral norms are extensively studied in literature, our understanding of structural norms is relatively nascent, and in the following section we will put forth one candidate proposal that suits this problem.

Clearly in this setup, constraining the spectral norm to be small allows all the matrices that would be allowed by the analogous structural constraint, and more. Thus it seems natural to hope that by moving to the more direct structural regularization, we might be able to obtain better results. As we will see in the following sections and the regret analysis, this intuition is true indeed.

3.2 Rank regularizing matrix norms

The constraint of low rank in the matrix domain is very similar to the notion of sparsity in the case of vectors. For vectors, it is well-known that minimizing the ℓ_1 norm leads to sparse solutions. Furthermore, as discussed in Chapter 11 of [6], regularization with an ℓ_p norm with p suitably close to 1 leads to the optimal regret bound up to constant factors in the vectorial setup. It is natural, therefore, to ask if one can define appropriate norms on matrices that regularize its rank to give low regret algorithms for online multitask learning. We will discuss two such norms below and define an algorithm and prove a regret bound for the latter.

The general scheme of online learning algorithms that we will be considering are algorithms of the form:

$$W_{t+1} = \text{argmin}_{W \in \Delta_{K \times N}} D(W, W_t) + \eta Tr(W^T X_t) \tag{5}$$

where D is the Bregman divergence induced by one of the norms to be described below. $\Delta_{K \times N}$ is the set of all $K \times N$ matrices which have all entries non-negative, and elements of each row add upto 1, i.e. form a distribution. Observe again that the projections are onto the space of all distributions over experts and not over a restricted subset due to the non-convexity of the rank constraint as explained above. Also note that we are using a linear loss here, which does not reduce the generality as it is well understood that curved loss functions only help the learner rather than the adversary in this setup.

3.3 Schatten L_p prenorms

Note that a rank m matrix has exactly m non-zero singular values in its singular value decomposition (SVD). Hence, doing an ℓ_p regularization on the vector of singular values with p close to 1 can be hoped to enforce only several non-zero singular values, leading to low rank matrices. Formally, define:

$$\|W\|_{S_p}^2 = \left(\sum_{i=1}^{r} |\sigma_i|^p \right)^{2/p} = Tr((WW^\top)^{p/2})^{2/p} \tag{6}$$

where $r = \min\{K, N\}$ and σ_i are the singular values of W. The second equality follows from the well-known fact that the eigenvalues of $WW^\top$ are the squared singular values of W. This is the norm used by Argyriou et al[4], however in the stochastic setup, and no theoretical analysis is provided.

Recently Cavallanti et al[5] carried out an analysis with these norms for a closely related multi-view problem. It is conceivable that a similar analysis extends to the multitask problem too, but is not discussed in their paper. It turns out however, that there is another norm much more conducive to analysis for this problem, and suited better to our problem intuition as argued below, which will be the main object of study in this paper.

3.4 Matrix (r, p) norms for structural regularization

For a $K \times N$ matrix W, let W^i and W_j refer to the i_{th} column and j_{th} row resp. of the matrix. The (r, p) norm of a matrix is given as:

$$\|W\|_{r,p}^2 = \left(\sum_{i=1}^{N} \left\|W^i\right\|_r^p \right)^{2/p} \tag{7}$$

It is easy to show that the above definition is indeed a norm on the space of matrices.

Lemma 1. *$\|\cdot\|_{r,p}$ is a norm on the space of matrices for $r, p \geq 1$.*

Proof. It is clear from the definition that $\|W\|_{r,p} \geq 0$ and is 0 iff $W \equiv 0$. Thus we only need to verify the triangle inequality.

Using the triangle inequality on the r norm,

$$\begin{aligned}\|W+U\|_{r,p} &= \left(\sum_{j=1}^{N} \left\|W^j + U^j\right\|_r^p \right)^{1/p} \\ &\leq \left(\sum_{j=1}^{N} \left(\left\|W^j\right\|_r + \left\|U^j\right\|_r\right)^p \right)^{1/p}.\end{aligned}$$

This term can now be seen as the p norm of a sum of two vectors of length N each with the j_{th} entry of one vector being $\left\|W^j\right\|_r$ and the second being $\left\|U^j\right\|_r$. Then by the triangle inequality of p norms on these two vectors we get

$$\begin{aligned}\|W+U\|_{r,p} &\leq \left(\sum_{j=1}^{N} \left\|W^j\right\|_r^p \right)^{1/p} + \left(\sum_{j=1}^{N} \left\|U^j\right\|_r^p \right)^{1/p} \\ &= \|W\|_{r,p} + \|U\|_{r,p}\end{aligned}$$

which gives us the triangle inequality, Thus $\|\cdot\|_{r,p}$ is indeed a norm on the space of matrices. □

It should be noted that this norm generalizes the (2,1) norm of [3].

The first thing to note is that except for specific choices of r, p, this is not a spectral norm in general. This norm directly acts on the entries of the matrix and can be different for two different matrices with the same eigenvalues. However, it does enforce the right structural properties on the matrix as explained below.

The r, p norm is a natural generalization of the ℓ_p norms to matrices, where we use r norm on columns, and then take a p norm of these values. To see why this is intuitive, consider the case of 0-1 matrices, with $r = \infty$ and $p = 1$. Then the norm of a column is 1 if it has at least 1 non-zero entry, 0 otherwise. Taking an ℓ_1 norm of these values corresponds to counting the number of experts that are being used. While competing with comparator in $\mathcal{S}_{\leq m}$, this is exactly the quantity we want to keep below m. Hence this norm does seem to capture the right intuition in our problem. That this is not a spectral norm might make it look less attractive on the first glance, but makes it much more amenable to analysis.

It might thus seem that using the aforementioned values of r and p would lead to the optimal regret bound. However, both these values are not suitable for analysis. Indeed, the ℓ_1 norm is not strictly convex, and, furthermore, our analysis requires $r, p \leq 2$. Thus we leave the choice of these exponents open for now, and hope to tune them to obtain the optimal bound once the analysis is complete.

An important property of r, p norms is also that for carefully tuned values of r, p they give a smaller norm to the matrices in $\mathcal{S}_{\leq m}$ than the matrices outside. This is crucial to our algorithm. Note that in Equation 5, we only project on the space of distribution matrices $\Delta_{K \times N}$. This means our learner can in general be outside $\mathcal{S}_{\leq m}$, which essentially means that the problem structure is not being adequately exploited, and we cannot hope for a significantly lower regret than solving the tasks indpendently in the worst case. However, if matrices in the set $\mathcal{S}_{\leq m}$ have a small value of the norm, then it is easy to show that the projection has a much greater chance of landing inside this set than outside. This property of the norms plays a crucial role in ensuring a low regret of this learning procedure.

4 Regret analysis

Let $L_T(U)$ for a matrix U denote $\sum_{t=1}^{T} Tr(U^T X_t)$, the cumulative loss using U for prediction. Also we use $L_T = \sum_{t=1}^{T} Tr(W_t^T X_t)$ to be the cumulative loss of our algorithm. For any matrix U, regret $R_T(U)$ with respect to U is defined by $L_T - L_T(U)$.

We will begin by stating the main result of this article, which will then be proved using a series of smaller lemmas.

Theorem 1. *Consider the learner using (5) with the D the Bregman divergence defined with respect to the (r,p) norms. Let L_T be its cumulative loss after T steps. Suppose that there are N experts and K tasks. Let $1 < p < r \leq 2$, and the loss of any expert at any time step be bounded by κ. Then for all $U \in \mathcal{S}_{\leq m}$, all T and all $\eta > 0$:*

$$L_T \leq L_T(U) + \frac{1}{2\eta} m^{2/p-2/r} K^{2/r} + \kappa T \eta \left(\frac{r}{r-1} + \frac{p}{p-1} - 2 \right) N^{2(p-1)/p} \tag{8}$$

The first thing we need to obtain in order to use this norm in the algorithm of (5) is to derive the Bregman divergence induced by this norm. It suffices to find the dual norm for this purpose. It turns out that the dual of (r,p) norm is the (s,q) norm where s and q are the exponents dual to r and p respectively. We begin by proving a Hölder's inequality for this norm.

Lemma 2. *Consider two matrices A and B, each of size $K \times N$. Then we have:*

$$|Tr(A^\top B)| \leq \|A\|_{r,p} \|B\|_{s,q} \tag{9}$$

where s and q are conjugate to r and p, respectively.

Proof. The result follows from a simple application of Hölder's inequality for vectors.

$$\begin{aligned}
|Tr(A^\top B)| &= \left| \sum_{j=1}^{N} {A^j}^T B^j \right| \\
&\leq \sum_{j=1}^{N} \left| {A^j}^T B^j \right| \leq \sum_{j=1}^{N} \left\| A^j \right\|_r \left\| B^j \right\|_s \qquad (10) \\
&\text{(Using Hölder's inequality for vectors on each element of the sum)} \\
&\leq \left(\sum_{j=1}^{N} \left\| A^j \right\|_r^p \right)^{1/p} \left(\sum_{j=1}^{N} \left\| B^j \right\|_s^q \right)^{1/q} \qquad (11) \\
&\text{(Hölder's inequality on vector of norms)} \\
&= \|A\|_{r,p} \|B\|_{s,q}
\end{aligned}$$

□

We can now simply derive the dual as indicated earlier.

Lemma 3. *Let $F(A) = 1/2 \|A\|_{r,p}^2$. Then its Legendre-Fenchel dual is given by $F^*(B) = 1/2 \|B\|_{s,q}^2$*

Proof. The dual function is defined as:

$$\begin{aligned}
F^*(B) &= \sup_A \left\{ Tr(A^T B) - \frac{1}{2} \|A\|_{r,p}^2 \right\} \\
&\leq \sup_A \left\{ \|A\|_{r,p} \|B\|_{s,q} - \frac{1}{2} \|A\|_{r,p}^2 \right\} \qquad (12) \\
&= \sup_A \left\{ \frac{1}{2} \|B\|_{s,q}^2 - \frac{1}{2} \left(\|B\|_{s,q} - \|A\|_{r,p} \right)^2 \right\} \qquad (13)
\end{aligned}$$

where the inequality follows from Lemma 2. Consider a particular choice of A in (13):

$$A_{ij} = B_{ij}^{(s-1)} \left\|B^j\right\|_s^{(q-s)} \|B\|_{s,q}^{(2-q)}. \tag{14}$$

We claim that this choice a) achieves the supremum in (13) and b) turns the inequality leading to (12) into an *equality*. If both of these points are verified, the statement of this lemma would follow. Let us start by showing a).

$$\begin{aligned}
\|A\|_{r,p} &= \left(\sum_{j=1}^{N} \left\|A^j\right\|_r^p\right)^{1/p} \\
&= \left(\sum_{j=1}^{N} \left(\sum_{i=1}^{K} \left(B_{ij}^{s-1} \left\|B^j\right\|_s^{(q-s)} \|B\|_{s,q}^{(2-q)}\right)^r\right)^{p/r}\right)^{1/p}
\end{aligned}$$

Using conjugacy of p, q and r, s, we see that $p = \frac{q}{q-1}$ and $r = \frac{s}{s-1}$. Substituting this above, we obtain

$$\begin{aligned}
\|A\|_{r,p} &= \left(\sum_{j=1}^{N} \left(\sum_{i=1}^{K} \left(B_{ij}^{s-1} \left\|B^j\right\|_s^{(q-s)} \|B\|_{s,q}^{(2-q)}\right)^{\frac{s}{s-1}}\right)^{q(s-1)/s(q-1)}\right)^{(q-1)/q} \\
&= \|B\|_{s,q}^{2-q} \left(\sum_{j=1}^{N} \left\|B^j\right\|_s^{(q-s)q/(q-1)} \left(\sum_{i=1}^{K} B_{ij}^{s}\right)^{q(s-1)/s(q-1)}\right)^{(q-1)/q} \\
&= \|B\|_{s,q}^{2-q} \left(\sum_{j=1}^{N} \left\|B^j\right\|_s^{(q-s)q/(q-1)} \left\|B^j\right\|_s^{q(s-1)/(q-1)}\right)^{(q-1)/q} \\
&= \|B\|_{s,q}^{2-q} \left(\sum_{j=1}^{N} \left\|B^j\right\|_s^{q}\right)^{(q-1)/q} \\
&= \|B\|_{s,q}^{2-q} \|B\|_{s,q}^{q-1} \\
&= \|B\|_{s,q}.
\end{aligned}$$

Hence, if A is defined as in (14), the non-negative second term of (13) vanishes, yielding $F^*(B) \leq \frac{1}{2}\|B\|_{s,q}^2$ and verifying a).

Observe that b) amounts to showing tightness of two inequalities in Lemma 2. Consider the quantity

$$\begin{aligned}
\left\|A^j\right\|_r^p &= \left\| B^{j\,(s-1)} \left\|B^j\right\|_s^{(q-s)} \|B\|_{s,q}^{(2-q)} \right\|_r^p \\
&= \|B\|_{s,q}^{p(2-q)} \left\|B^j\right\|_s^{p(q-s)} \left(\sum_{i=1}^K B_{ij}^{r(s-1)}\right)^{p/r} \\
&= \|B\|_{s,q}^{p(2-q)} \left\|B^j\right\|_s^{p(q-s)} \left(\sum_{i=1}^K B_{ij}^{s}\right)^{p(s-1)/s} \\
&= \|B\|_{s,q}^{p(2-q)} \left\|B^j\right\|_s^{p(q-s)} \left\|B^j\right\|_s^{p(s-1)} \\
&= \|B\|_{s,q}^{p(2-q)} \left\|B^j\right\|_s^{p(q-1)} \qquad (*) \\
&= \|B\|_{s,q}^{p(2-q)} \left\|B^j\right\|_s^{q} \\
&= \left\|B^j\right\|_s^q \|B\|_{s,q}^p \|B\|_{s,q}^{q(1-q)/(q-1)} \\
&= \frac{\left\|B^j\right\|_s^q}{\|B\|_{s,q}^q} \|A\|_{r,p}^p
\end{aligned}$$

This means that $\frac{\left\|A^j\right\|_r^p}{\|A\|_{r,p}^p} = \frac{\left\|B^j\right\|_s^q}{\|B\|_{s,q}^q}$ which makes the application of Hölder's inequality in (11) tight. Furthermore, using an intermediate point $(*)$,

$$\begin{aligned}
\frac{A_{ij}^r}{\left\|A^j\right\|_r^r} &= \frac{\left(B_{ij}^{(s-1)} \left\|B^j\right\|_s^{(q-s)} \|B\|_{s,q}^{(2-q)}\right)^r}{\left\|A^j\right\|_r^r} \\
&= \frac{B_{ij}^{r(s-1)} \left\|B^j\right\|_s^{r(q-s)} \|B\|_{s,q}^{r(2-q)}}{\|B^j\|_s^{(q-1)r} \|B\|_{s,q}^{r(2-q)}} \\
&= \frac{B_{ij}^s}{\left\|B^j\right\|_s^s}
\end{aligned}$$

which makes the inequality in (10) tight as well. Thus this choice of A yields the desired result, that is $F^*(B) = \frac{1}{2}\|B\|_{s,q}^2$ completing the proof. □

The above proof gives us the value of the matrix A at which the maximum in the expression for the dual is attained. It is worthwhile to spend a minute inspecting the mapping between primal and dual spaces obtained above. It will be shown that the values of q, s that yield a good regret have $q \gg s$. For such values of q, the mapping from dual space to primal space tries to concentrate most of the mass of a row in the columns having largest entries in the dual matrix B. Thus in mapping back to the space of weight matrices, our norm implicitly tries to minimize the number of non-zero columns which is also the number of experts used. This further supports our intuition that this norm is well-suited for the problem.

Note that it is well known that if A attains the supremum in the definition of $F^*(B)$, then A and B form a primal-dual pair, with $A = \nabla F^*(B)$ and $B = \nabla F(A)$. So, in particular, we now have the derivative of our norm, $\nabla_A \frac{1}{2}\|A\|_{r,p}^2 = A_{ij}^{(r-1)} \left\|A^j\right\|_r^{(p-r)} \|A\|_{r,p}^{(2-p)}$. We can now define a Bregman divergence using this norm as the Bregman function. We have

$$D_{r,p}(W_{t-1}, W_t) = \frac{1}{2}\|W_{t-1}\|_{r,p}^2 - \frac{1}{2}\|W_t\|_{r,p}^2 - Tr({V_t}^T(W_{t-1} - W_t)), \tag{15}$$

where V_t is the dual image of W_t as defined above, and $D_{r,p}$ is the Bregman divergence using $\|\cdot\|_{r,p}^2$ as the Bregman function.

As the last preliminary result, we deduce an upper bound on the (r,p)-norms that will be useful in the later analysis.

Lemma 4. *Let $1 < p < r \leq 2$. Then $\forall U \in \mathcal{S}_{\leq m}$, $\|U\|_{r,p} \leq K^{1/r} m^{1/p-1/r}$.*

Proof. The fact that $U \in \mathcal{S}_{\leq m}$ implies that it has at most m non-zero columns. Also, we have to pick at least one expert for each task.

First note that it suffices to look at just 0-1 comparator matrices, as the adversary will always pick the set of m experts and task assignments to those experts that result in the smallest overall loss over the entire sequence. With that in mind, we can set up the following optimization problem:

$$\max_{n_1,\ldots,n_m \geq 0} \left(n_1^{p/r} + n_2^{p/r} + \ldots n_m^{p/r}\right)^{1/p}$$
$$\text{s.t.} \quad n_1 + \ldots n_m = K$$

It is easy to show using a quick second derivative computation that this objective is a concave function of the n_i's. So, we can compute the Lagrange function, and set its derivative to zero, which gives us:

$$\frac{1}{r} n_i^{p/r-1} \left(\sum_i n_i^{p/r}\right)^{(1-p)/p} = \lambda$$

where λ is the Lagrange multiplier, for all $i = 1 \ldots m$. This means that all the n_i's are equal to K/m. Evaluating the norm using a matrix U with m non-zero columns, each having K/m ones yields the desired result. □

We are now in a position to prove the theorem.

Proof of Theorem 1.

For predictors of the form (5), the regret can be bounded as:

$$R_T(U) \leq \frac{1}{\eta} D_{r,p}(U, W_0) + \frac{1}{\eta} \sum_{t=1}^{T} D_{r,p}(W_{t-1}, W_t) \tag{16}$$

This form of regret bound is well-known, for example Lemma 10 in the unpublished lecture notes [9] as well as [6] . Thus the key step of the theorem is to bound the two divergence terms $D_{r,p}(U, W_0)$ and $D_{r,p}(W_{t-1}, W_t)$.

Note that we can take W_0 to be uniform (i.e. $1/N$ over experts for each task). Then the entries of V_0 are uniform too, i.e., $V_{0ij} = W_{0ij}^{(r-1)} \left\|W_0^j\right\|_r^{(p-r)} \|W_0\|_{r,p}^{(2-p)} = c$ for all i, j as the column norms W_0^j are same for all columns for some constant c. So we have

$$\begin{aligned} Tr(V_0{}^T(U - W_0)) &= \sum_{i,j} V_{0ij}(U_{ij} - W_{0ij}) \\ &= c \sum_{i,j} \left(U_{ij} - W_{0ij}\right) \\ &= 0 \quad \text{(as each row of both } U \text{ and } W_0 \text{ add up to 1)} \end{aligned}$$

So the linear term of the first Bregman divergence term in (16) is zero, and hence this divergence is large when the comparator matrix U has a large (r, p) norm. Using Lemma 4 this happens precisely when the matrix U has exactly m non-zero columns, each with K/m ones. Note that this follows from the assumption that each row of the matrix U forms a distribution and hence sums to 1. In this case, the norm of U is $m^{1/p-1/r} K^{1/r}$. Dropping the negative $\|W_0\|_{r,p}$ term, the first divergence term is upper bounded as

$$D_{r,p}(U, W_0) \leq \frac{1}{2}\|U\|_{r,p}^2 \leq \frac{1}{2} m^{2/p-2/r} K^{2/r}$$

for all $U \in \mathcal{S}_{\leq m}$.

We next turn to the divergences between iterates, $D_{r,p}(W_{t-1}, W_t)$. For convenience, define $\tilde{W}_{t+1}$ as follows:

$$\tilde{W}_{t+1} = \text{argmin}_W D_{r,p}(W, W_t) + \eta Tr(W^T X_t) \tag{17}$$

i.e. the unconstrained minimizer of the optimization problem. Then it is easily shown that $W_{t+1} = \Pi_{r,p}(W_t; \Delta_{K\times N})$, where $\Pi_{r,p}(W; S)$ is the Bregman projection of a matrix W onto the set S, using the Bregman function $\frac{1}{2}\|\cdot\|^2_{r,p}$.

Also, using the Pythagorean inequality for Bregman divergences (see for example [6] Lemma 11.3), we can write:

$$\begin{aligned} D_{r,p}(W_{t-1}, \tilde{W}_t) &\geq D_{r,p}(W_{t-1}, W_t) + D_{r,p}(W_t, \tilde{W}_t) \\ &\geq D_{r,p}(W_{t-1}, W_t) \end{aligned} \tag{18}$$

This means that we can bound the regret further from (16) as:

$$R_T(U) \leq \frac{1}{\eta} D_{r,p}(U, W_0) + \frac{1}{\eta} \sum_{t=1}^{T} D_{r,p}(W_{t-1}, \tilde{W}_t) \tag{19}$$

Now, if $\tilde{V}_t$ is the dual variable corresponding to $\tilde{W}_t$, we can easily argue that $\tilde{V}_t = V_{t-1} + \eta X_t$. This can be seen by differentiating the objective in (17) and setting it to zero. It is a property of Bregman divergences (Prop. 11.1 of [6]) that $D_f(W_{t-1}, \tilde{W}_t) = D_{f^*}(\tilde{V}_t, V_{t-1})$ where f^* is the convex conjugate of f and V_t is the conjugate dual of W_t, specified by the gradient mapping $V_t = \nabla f(W_t)$. In the particular context of this problem, this property is relevant as the update equations from (5) are very simple in the dual space. The dual updates can be written as $\tilde{V}_t = V_{t-1} + \eta X_{t-1}$ which is simply the gradient condition at optimality from the fact that the derivative of $D_f(W, W_t)$ wrt W is simply $\nabla f(W) - \nabla f(W_t)$.

Using Lemma 5 that we will prove below, these divergences are bounded as:

$$D_{s,q}(\tilde{V}_t, V_{t-1}) \leq \eta^2 (s + q - 2) \|X_{t-1}\|^2_{s,q} \tag{20}$$

Assuming that the losses are componentwise bounded at each time, and noting that we have non-zero loss for exactly one task, we can bound $\|X_t\|^2_{s,q}$ with $\kappa N^{2/q}$, where κ is a bound on each entry of X_t uniformly across $t = 1 \ldots T$. Summing terms over time, and using the fact that $s = \frac{r}{r-1}$ and $q = \frac{p}{p-1}$ completes the proof.

□

Lemma 5. *Let $W_t, \tilde{W}_{t+1}$ be as in (17), with divergence induced by the (r, p) norm ,with $1 \leq r, p \leq 2$. Let V_t and $\tilde{V}_{t+1}$ be the corresponding dual images, and s, q be the dual exponents to r, p resp. Then we have:*

$$D_{s,q}(\tilde{V}_{t+1}, V_t) \leq \eta^2 (s + q - 2) \|X_t\|^2_{s,q} \tag{21}$$

Proof. The key idea as in most proofs of this kind is to use the fact that Bregman divergence measures the difference between a function and its first order Taylor approximation. We know that this difference is equal to the second order Taylor term at some intermediate point by the mean value theorem. So if we can uniformly bound the Hessian matrix of our regularizer, it suffices to demonstrate a bound on the divergences. Let $F(V) = \frac{1}{2}\|V\|^2_{s,q}$. The Hessian matrix for this function is given by $H_{(ij,kl)} = \frac{\partial F(V)}{\partial V_{ij} \partial V_{kl}}$. We can think of the Hessian as either a 4-dimensional matrix, or, as a 2-dimensional one. The latter is obtained by letting the index ij stand for $K \cdot (i-1) + j$. With this notation, the second order term in the Mean Value Theorem is

$$\frac{1}{2} (\text{vec}(V_t - \tilde{V}_{t+1}))^\top H(\bar{V}) (\text{vec}(V_t - \tilde{V}_{t+1})). \tag{22}$$

Here vec is an operator that stretches out its matrix arguments to a vector and $\bar{V} = \alpha V_t + (1 - \alpha)\tilde{V}_{t+1}$ for some $\alpha \in [0, 1]$. Written as a summation, this is simply $\sum_{i,j,k,l} (V_t - \tilde{V}_{t+1})_{ij} H(\bar{V})_{ij,kl} (V_t - \tilde{V}_{t+1})_{kl}$. We now use the fact that $\tilde{V}_{t+1} = V_t + \eta X_t$ to write this as $\eta^2 \sum_{i,j,k,l} (X_t)_{ij} H(\bar{V})_{ij,kl} (X_t)_{kl}$.

We now drop the subscript t and the bar from $\bar{V}$ to ease the notation a bit. Let us look at a particular entry of the matrix H. By applying the chain rule,

$$\begin{aligned}
H_{ij,kl} &= \frac{\partial F(V)}{\partial V_{ij}\partial V_{kl}} \\
&= \frac{\partial \left(\nabla_V F(V)\right)_{kl}}{\partial V_{ij}} \\
&= \frac{\partial \|V\|_{s,q}^{(2-q)} \left\|V^l\right\|_s^{(q-s)} V_{kl}^{(s-1)}}{\partial V_{ij}} \\
&= \frac{\partial \|V\|_{s,q}^{(2-q)}}{\partial V_{ij}} \left\|V^l\right\|_s^{(q-s)} V_{kl}^{(s-1)} \\
&+ \frac{\partial \left\|V^l\right\|_s^{(q-s)}}{\partial V_{ij}} \|V\|_{s,q}^{(2-q)} V_{kl}^{(s-1)} \\
&+ \frac{\partial V_{kl}^{(s-1)}}{\partial V_{ij}} \|V\|_{s,q}^{(2-q)} \left\|V^l\right\|_s^{(q-s)}
\end{aligned}$$

Considering each of the three terms above, we get

$$\begin{aligned}
H_{ij,kl} &= \frac{(2-q)}{q} \|V\|_{s,q}^{(2-2q)} \frac{q}{s} \left\|V^j\right\|_s^{(q-s)} s V_{ij}^{(s-1)} \left\|V^l\right\|_s^{(q-s)} V_{kl}^{(s-1)} \\
&+ \mathbb{I}(j=l) \|V\|_{s,q}^{(2-q)} \frac{(q-s)}{s} \left\|V^l\right\|_s^{(q-2s)} s V_{il}^{(s-1)} V_{kl}^{(s-1)} \\
&+ \mathbb{I}(i=k, j=l) \|V\|_{s,q}^{(2-q)} \left\|V^l\right\|_s^{(q-s)} (s-1) V_{kl}^{(s-2)} \\
&= \underbrace{(2-q)\|V\|_{s,q}^{(2-2q)} \left\|V^j\right\|_s^{(q-s)} \left\|V^l\right\|_s^{(q-s)} V_{ij}^{(s-1)} V_{kl}^{(s-1)}}_{A} \\
&+ \underbrace{\mathbb{I}(j=l)(q-1)\|V\|_{s,q}^{(2-q)} \left\|V^j\right\|_s^{(q-2)} \left\|V^j\right\|_s^{2(1-s)} V_{ij}^{(s-1)} V_{kj}^{(s-1)}}_{B} \\
&+ \underbrace{\mathbb{I}(j=l)(1-s)\|V\|_{s,q}^{(2-q)} \left\|V^j\right\|_s^{(q-1)} \left\|V^j\right\|_s^{(1-2s)} V_{ij}^{(s-1)} V_{kj}^{(s-1)}}_{C} \\
&+ \underbrace{\mathbb{I}(j=l, i=k)(s-1)\|V\|_{s,q}^{(2-q)} \left\|V^j\right\|_s^{(q-s)} V_{ij}^{(s-2)}}_{D}.
\end{aligned}$$

Now let us consider the summation $\sum_{i,j,k,l} H_{ij,kl} X_{ij} X_{kl}$ by looking at the contribution of each of the terms A, B, C, D separately.

For the term A, we see that

$$\begin{aligned}
&\sum_{ij,kl} X_{ij} X_{kl} (2-q) \|V\|_{s,q}^{(2-2q)} \left\|V^j\right\|_s^{(q-s)} \left\|V^l\right\|_s^{(q-s)} V_{ij}^{(s-1)} V_{kl}^{(s-1)} \\
&= (2-q)\|V\|_{s,q}^{(2-2q)} \left(\sum_{j=1}^{N} \left\|V^j\right\|_s^{(q-s)} \sum_{i=1}^{K} V_{ij}^{(s-1)} X_{ij} \right)^2 \\
&\leq 0,
\end{aligned}$$

as $q \geq 2$ by the assumption that $p \leq 2$.

A similar argument shows that the contribution of term C is negative as well, and hence these two terms can be ignored for purposes of getting upper bounds. So in order to show an upper bound, we just need to account for the contributions of the terms B and D.

Consider the contribution of the term B. Excluding the leading η^2, the sum over these terms can be written as:

$$\frac{1}{\|V\|_{s,q}^{(q-2)}} \sum_{j=1}^{N} (q-1) \left\|V^j\right\|_s^{q-2} \left(\frac{1}{(\sum_{i=1}^K V_{ij}^s)^{(s-1)/s}} \right)^2 \sum_{i,k=1}^{K} X_{ij} X_{kj} V_{ij}^{(r-1)} V_{kj}^{(r-1)}$$

$$= \frac{1}{\|V\|_{s,q}^{(q-2)}} \sum_{j=1}^{N} (q-1) \left\|V^j\right\|_s^{q-2} \left(\frac{\sum_{i=1}^K X_{ij} V_{ij}^{(r-1)}}{(\sum_{i=1}^K V_{ij}^s)^{(s-1)/s}} \right)^2$$

$$\leq \frac{(q-1)}{\cancel{\|V\|_{s,q}^{(q-2)}}} \cancel{\left(\sum_{j=1}^{n} \left\|V^j\right\|_s^q \right)^{(q-2)/q}} \left(\sum_{j=1}^{N} \left(\frac{\sum_{i=1}^K X_{ij} V_{ij}^{(r-1)}}{(\sum_{i=1}^K V_{ij}^s)^{(s-1)/s}} \right)^{2q/2} \right)^{2/q}$$

Using Hölder's inequality with exponents $\frac{q}{q-2}$ and $\frac{q}{2}$

$$\leq (q-1) \left(\sum_{j=1}^{N} \left(\frac{\cancel{\left(\sum_{i=1}^K V_{ij}^{(s-1)s/(s-1)}\right)}}{\cancel{(\sum_{i=1}^K V_{ij}^s)^{(s-1)/s}}} \left(\sum_{i=1}^{K} X_{ij}^s \right)^{1/s} \right)^q \right)^{2/q}$$

Using Hölder's inequality with exponents $\frac{s}{s-1}$ and s

$$= (q-1)\|X\|_{s,q}^2$$

Now we look at the second term. Using similar applications of Hölder's inequality with slightly different exponents, we can bound the terms where $i = k, j = l$ by $(s-1)\|X\|_{s,q}^2$. The other two cases have been shown to be negative and are thus dropped from the upper bound. Adding all the terms gives us the desired upper bound. □

Note that when $q > s$, then the decomposition of the middle term into terms B and C in the above proof is not needed, and we can get a slightly better regret bound that involves just $(q-1)$ in place of $(s+q-2)$. However, this doesn't cause any significant change in the bound unless s is very large, and hence we use the slightly loose but more general form above for further discussion.

4.1 Optimal setting of parameters

As the reader would have observed, the applications of Hölder's inequalities in the previous steps critically relied on the fact that $s \geq 2, q \geq 2$. Thus we need to have $1 < r, p \leq 2$. From the bound of (8), it is clear that the best setting of η is to balance the two terms. This allows us to rewrite the bound as:

$$R_T(U) \leq m^{1/p-1/r} K^{1/r} N^{(p-1)/p} \sqrt{\kappa T \left(\frac{1}{r-1} + \frac{1}{p-1} \right)} \tag{23}$$

It is not obvious what the optimal setting for r, p is in general. However, we can investigate certain regimes in which it is possible to set the values of r, p in a way that brings our regret bound very close to the optimal regret of $O\left(\sqrt{T\left(K \log m + m \log N\right)}\right)$.

Consider setting $K = N^\alpha$ for some $\alpha > 0$. Then the optimal regret bound is dominated by the term $O\left(\sqrt{T \log m} N^{\alpha/2}\right)$. If we plug this value of K in our regret bound, our regret scales as

$$O\left(m^{1/p-1/r} N^{\alpha/r+(p-1)/p} \sqrt{T \left(\frac{1}{r-1} + \frac{1}{p-1} \right)} \right).$$

Optimizing over the exponent of N results in the choice $r = \frac{2\alpha p}{\alpha(p-1)+1}$ which is between 1 and 2 only when $\alpha \leq 1$. However, multitask problems are interesting when the number of tasks is very large, potentially much larger than the number of experts themselves, so that performing the tasks independently is really bad.

Consider the case $\alpha > 2$. If we set $r = 2$ and $\frac{1}{p-1} = \log m$, our regret scales as

$$O\left(m^{-1/2} N^{\left(\alpha/2 + \frac{1}{(1+\log m)}\right)} \sqrt{T(1+\log m)}\right)$$

which is very close to the optimal bound.

4.2 Comparison with existing results

The previous best results on this problem are in the paper of Cavallanti et al[5]. This paper describes a multi-task p-norm perceptron algorithm, for which a mistake bound is shown. To compare our regret bound with their mistake bound, we first need to measure the two algorithms under the same loss function. For this, we first state an easy reduction to a hinge loss regret bound for any algorithm that gives bounded regret in the experts setup.

Lemma 6. *Consider any online algorithm that takes a sequence of losses x_t on experts and outputs a distribution over them with the cumulative regret bounded as $R(T)$. Then there is an algorithm that has its regret under hinge loss bounded by $R(T)$ in any classification problem when compared to all possible weight vectors in the probability simplex.*

Proof. The proof is a simple reduction that uses the experts algorithm as a black box. Suppose at time t, our algorithm has a distribution w_t over the experts. We receive a query point x_{t+1} and make a prediction $\text{sign}(w_t^\top x_{t+1})$. Then we receive y_{t+1}. If our prediction is correct, then we pass a loss vector of all zeros to our algorithm, otherwise we pass the query point $-y_{t+1}x_{t+1}$ to it. The regret bound of our algorithm implies that

$$\sum_{t=1}^{T} -y_{t+1} w_t^\top x_{t+1} \mathbb{I}(y_{t+1} w_t^\top x_{t+1} < 1) \leq \sum_{t=1}^{T} -y_{t+1} u^\top x_t \mathbb{I}(y_{t+1} w_t^\top x_{t+1} < 1) + R(T)$$

for any distribution u over the experts. Adding $\sum_{t=1}^{T} \mathbb{I}(y_{t+1} w_t^\top x_{t+1} < 1)$ to both sides gives

$$\sum_{t=1}^{T} (1 - y_{t+1} w_t^\top x_{t+1}) \mathbb{I}(y_{t+1} w_t^\top x_{t+1} < 1) \leq \sum_{t=1}^{T} (1 - y_{t+1} u^\top x_t) \mathbb{I}(y_{t+1} w_t^\top x_{t+1} < 1) + R(T)$$

The left hand side is the hinge loss of our algorithm, while the right hand side is an upper bound on the hinge loss of the comparator, which completes the proof. □

The reason why this lemma is useful is that we can directly translate our regret bound to a regret bound under hinge loss in a classification setup, thus allowing direct comparison to the results of [5]. The regret bound in that paper scales as $O((K-m)\sqrt{T \log \max\{K, N\}})$. Putting $K = N^\alpha$, we see that this regret scales as $O(N^\alpha)$ which is much worse than a near optimal regret of roughly $O(N^{\alpha/2})$ achieved by our algorithm.

5 Conclusion

In this paper we have examined a multitask online learning problem. The key challenge in this problem is that the learner needs to infer task relatedness along the way. We cast this as a matrix regularization problem, which leads to two possible approaches based on structural and spectral regularization. We work with a structural matrix norm, that leads to computationally efficient low regret algorithms. The regret bounds we obtained are not optimal, but, as we demonstrated, get quite close to the optimal regret for some settings of problem parameters. The bounds seem to be the best known bounds for any deterministic and computationally feasible algorithm for this problem.

References

[1] Jacob Abernethy, Peter L. Bartlett, and Alexander Rakhlin. Multitask learning with expert advice. In *Proceedings of the Conference on Learning Theory*, pages 484–498, 2007.

[2] Yonatan Amit, Michael Fink, Nathan Srebro, and Shimon Ullman. Uncovering shared structures in multiclass classification. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 17–24, New York, NY, USA, 2007. ACM.

[3] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 2008.

[4] Andreas Argyriou, Charles A. Micchelli, Massimiliano Pontil, and Yiming Ying. A spectral regularization framework for multi-task structure learning. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 25–32. MIT Press, Cambridge, MA, 2008.

[5] Giovanni Cavallanti, Nicoló Cesa-Bianchi, and Claudio Gentile. Linear algorithms for online multitask classification. In *Proceedings of Conference on Learning Theory*, 2008.

[6] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.

[7] Ofer Dekel, Philip M. Long, and Yoram Singer. Online learning of multiple tasks with a shared loss. *J. Mach. Learn. Res.*, 8:2233–2264, 2007.

[8] Brian Kulis, Mátyás Sustik, and Inderjit Dhillon. Learning low-rank kernel matrices. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 505–512, New York, NY, USA, 2006. ACM.

[9] Alexander Rakhlin. Lecture notes on online learning. *Unpublished lecture notes*, 2008.