Ultra-Fast Optimization Algorithm for Sparse Multi Kernel Learning

Francesco Orabona FRANCESCO@ORABONA.COM
DSI, Università degli Studi di Milano, Via Comelico 39, 20135 - Milano, Italy

Luo Jie JLUO@IDIAP.CH
Idiap Research Institute, Centre du Parc, 1920 - Martigny, Switzerland
École Polytechnique Fédérale de Lausanne (EPFL), 1015 - Lausanne, Switzerland

Abstract

Many state-of-the-art approaches for Multi Kernel Learning (MKL) struggle at finding a compromise between performance, sparsity of the solution and speed of the optimization process. In this paper we look at the MKL problem at the same time from a learning and optimization point of view. So, instead of designing a regularizer and then struggling to find an efficient method to minimize it, we design the regularizer while keeping the optimization algorithm in mind. Hence, we introduce a novel MKL formulation, which mixes elements of p-norm and elastic-net kind of regularization. We also propose a fast stochastic gradient descent method that solves the novel MKL formulation. We show theoretically and empirically that our method has 1) state-of-the-art performance on many classification tasks; 2) exact sparse solutions with a tunable level of sparsity; 3) a convergence rate bound that depends only logarithmically on the number of kernels used, and is independent of the sparsity required; 4) independence on the particular convex loss function used.

1. Introduction

In recent years there has been a lot of interest in designing principled classification algorithms over multiple cues, based on the intuitive notion that using more features should lead to better performance. Focusing to the domain of the Support vector machines (SVM) (Cristianini & Shawe-Taylor, 2000), the use of multiple cues has been translated in the use of multiple kernels, weighted by some positive coefficients.

A recent approach in this field is to use a two-stage procedure, in which the first stage finds the optimal weights to combine the kernels, using an improved definition of the kernel alignment (Cristianini et al., 2002) as a proxy of the generalization error, and a standard SVM as second stage (Cortes et al., 2010). However in this approach, even if theoretically principled, the global optimality is not guaranteed, because the optimization process split in two phases.

A different approach with a joint optimization process is Multi Kernel Learning (MKL) (Lanckriet et al., 2004; Rakotomamonjy et al., 2008; Sonnenburg et al., 2006; Nath et al., 2009; Zien & Ong, 2007). In MKL one solves a joint optimization problem while also learning the optimal weights for combining the kernels. MKL methods are theoretically founded, which are based on the minimization of an upper bound of the generalization error (Kakade et al., 2009; Cortes et al., 2010), like in standard SVM. However solving it is far more complex than training a single SVM classifier. The main difficulty lies in designing efficient optimization algorithms, especially when a sparse solution is wanted. Sparsity is often achieved using an l_1 norm as regularizer or as constraint. Unfortunately, the l_1 norm is not smooth, so it slows down the optimization process.

Most of proposed algorithms for MKL solve this difficult optimization problem with an alternating optimization approach, first optimizing over the kernel combination weights, with the current SVM solution fixed, then finding the SVM solution, given the current weights (Sonnenburg et al., 2006; Rakotomamonjy et al., 2008; Xu et al., 2008). One advantage of the alternating optimization approach is that it is possible to use existing efficient SVM solvers for the SVM optimization step. On the other hand, for

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

these algorithms, even if they are known to converge, it usually is not possible to prove a bound on the maximum number of iterations needed. For the same reason it is not possible to compute how the relevant quantities affect the asymptotic computational complexity, and often these dependencies are estimated numerically for the specific implementation at hand. For example, SILP multiclass MKL algorithm (Zien & Ong, 2007) seems to depend polynomially on the number of training examples and number of classes with an exponent of ~ 2.4 and ~ 1.7 respectively. For the other algorithms these dependencies are not clear. Moreover, the learning process is usually stopped early, based on the common assumption that it is enough to have an approximate solution. This approach can be dangerous when a dual algorithm is used, potentially stopping far away from the optimal solution (Chapelle, 2007).

Notable exception to the use of l_1 regularization for MKL are (Kloft et al., 2009; Orabona et al., 2010; Vishwanathan et al., 2010) in which a l_p norm constraint is introduced, to have a simpler problem and to be able to tune the level of "sparsity" of the solution. However in this case the true sparsity is lost, and the weights of the kernels, even if they can become extremely small, will never be exactly zero. Another limitation is that many of these algorithms relies on particular loss functions, and the entire algorithm has to be changed if the loss function is changed (Kloft et al., 2009; Vishwanathan et al., 2010).

In this paper we look at the same time at the MKL problem from a learning and optimization points of view. Therefore, instead of designing a regularizer and then try to find an efficient method to minimize it, we design the regularizer while keeping the optimization process in mind. In other words, a perfect regularizer is useless if it is impractical to be used. The novel MKL formulation that we propose gives 1) state-of-the-art performance on many classification tasks; 2) exact sparse solutions with a tunable level of sparsity; 3) a convergence rate bound that depends only logarithmically on the number of kernels used, and is independent of the sparsity required; 4) independence on the particular convex loss function used. As in (Orabona et al., 2010; Jie et al., 2010; Martins et al., 2011), our algorithm solves the optimization problem directly in the primal formulation. This allows us to use any complex loss functions, as the multiclass loss in (Crammer & Singer, 2002) or more in general structured losses (Tsochantaridis et al., 2004), with minimal changes to the algorithm. We call this algorithm Ultra Fast Online Multi Kernel Learning, UFO-MKL.

The rest of the paper presents the theory and the experimental results supporting our claims. Section 2 revises the basic definitions and the mathematical tools needed, Section 3 introduces the Multi Kernel Learning problem. Section 4 presents the theory and algorithm of UFO-MKL, while Section 5 reports experiments on binary and multiclass classification tasks.

2. Preliminaries

In this section we introduce formally the notation, and the needed mathematical tools. We indicate matrices and vectors with bold letters. We also indicate with a bar, e.g. $\bar{\boldsymbol{w}}$, the vector formed by the concatenation of the F vectors $\boldsymbol{w}^j$, hence $\bar{\boldsymbol{w}} = [\boldsymbol{w}^1, \boldsymbol{w}^2, \cdots, \boldsymbol{w}^F]$.

We also introduce some concepts of convex analysis that are needed in the following. For a more thorough introduction see for example (Boyd & Vandenberghe, 2004). Given a convex function $g : S \to \mathbb{R}$, its Fenchel conjugate $g^* : S \to \mathbb{R}$ is defined as $g^*(\boldsymbol{u}) = \sup_{\boldsymbol{v} \in S}(\boldsymbol{v}^T\boldsymbol{u} - g(\boldsymbol{v}))$. A vector $\boldsymbol{x}$ is a subgradient of a function g at $\boldsymbol{v}$, if $\forall \boldsymbol{u} \in S, g(\boldsymbol{u}) - g(\boldsymbol{v}) \geq (\boldsymbol{u} - \boldsymbol{v}) \cdot \boldsymbol{x}$. The differential set of g at $\boldsymbol{v}$, indicated with $\partial g(\boldsymbol{v})$, is the set of all the subgradients of g at $\boldsymbol{v}$. If g is convex and differentiable at $\boldsymbol{v}$ then $\partial g(\boldsymbol{v})$ consists of a single vector which is the gradient of g at $\boldsymbol{v}$ and is denoted by $\nabla g(\boldsymbol{v})$. A function $g : S \to \mathbb{R}$ is said to be λ-strongly convex w.r.t. a convex and differentiable function h iff for any $\boldsymbol{u}, \boldsymbol{v} \in S$ and any subgradient $\partial g(\boldsymbol{u})$, $g(\boldsymbol{v}) \geq g(\boldsymbol{u}) + \partial g(\boldsymbol{u}) \cdot (\boldsymbol{v} - \boldsymbol{u}) + \lambda(h(\boldsymbol{v}) - h(\boldsymbol{u}) - (\boldsymbol{v} - \boldsymbol{u}) \cdot \nabla h(\boldsymbol{v}))$, where the terms in parenthesis form the Bregman divergence of h between $\boldsymbol{v}$ and $\boldsymbol{u}$.

Let $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$, with $N \in \mathbb{N}$, $\boldsymbol{x}_i \in \mathbb{X}$ and $y_i \in \mathbb{Y}$, be the training set. Consider a function $\boldsymbol{\phi}(\boldsymbol{x}) : \mathbb{X} \to \mathbb{H}$ that maps the samples into a high, possibly infinite, dimensional space. In the binary case $\mathbb{Y} = \{-1, 1\}$, and we use the standard setting to learn with kernels[1], in which the prediction on a sample $\boldsymbol{x}$ is a function of the scalar product between an hyperplane $\boldsymbol{w}$ and the transformed sample $\boldsymbol{\phi}(\boldsymbol{x})$. With multiple kernels, we will have F corresponding functions $\boldsymbol{\phi}^j(\cdot)$, $j = 1, \cdots, F$, and F corresponding kernels $K^j(\boldsymbol{x}, \boldsymbol{x}')$ defined as $\boldsymbol{\phi}^j(\boldsymbol{x}) \cdot \boldsymbol{\phi}^j(\boldsymbol{x}')$.

For multiclass and structured classification $\mathbb{Y} = \{1, \ldots, M\}$, we follow the common approach to use joint feature maps $\boldsymbol{\phi}(\boldsymbol{x}, y) : \mathbb{X} \times \mathbb{Y} \to \mathbb{H}$ (Tsochantaridis et al., 2004). Again, we will have F functions $\boldsymbol{\phi}^j(\cdot, \cdot), j = 1, \cdots, F$, and F kernels $K^j((\boldsymbol{x}, y), (\boldsymbol{x}', y')) = \boldsymbol{\phi}^j(\boldsymbol{x}, y) \cdot \boldsymbol{\phi}^j(\boldsymbol{x}', y')$. This definition includes the case of training M different hyperplanes, one for each class. In fact $\boldsymbol{\phi}^j(\boldsymbol{x}, y)$ can be

[1] For simplicity we will not use the bias, it can be easily added modifying the kernel definition.

defined as

$$\boldsymbol{\phi}^j(\boldsymbol{x}, y) = [\mathbf{0}, \cdots, \mathbf{0}, \underbrace{\boldsymbol{\phi}'^j(\boldsymbol{x})}_{y}, \mathbf{0}, \cdots, \mathbf{0}],$$

where $\boldsymbol{\phi}'^j(\cdot)$ is a transformation that depends only on data. Similarly $\boldsymbol{w}$ will be composed by M blocks, $[\boldsymbol{w}^1, \cdots, \boldsymbol{w}^M]$. According to the defined notation, $\bar{\boldsymbol{\phi}}(x, y) = [\boldsymbol{\phi}^1(x, y), \cdots, \boldsymbol{\phi}^F(x, y)]$. With a slight abuse of notation, in the following we will denote by $\bar{\boldsymbol{\phi}}(\boldsymbol{x}, \cdot)$ both the binary and multiclass feature transform.

A $(2, p)$ *group norm* $\|\bar{\boldsymbol{w}}\|_{2,p}^2$ on $\bar{\boldsymbol{w}}$ is defined as

$$\|\bar{\boldsymbol{w}}\|_{2,p} := \left\| \left[\|\boldsymbol{w}^1\|_2, \|\boldsymbol{w}^2\|_2, \cdots, \|\boldsymbol{w}^F\|_2 \right] \right\|_p,$$

that is the p-norm of the vector of F elements, formed by 2-norms of the vectors $\boldsymbol{w}^j$. The dual norm of $\|\cdot\|_{2,p}$ is $\|\cdot\|_{2,q}$, where $1/p + 1/q = 1$ (Kakade et al., 2009).

3. Multi Kernel Learning and Regularizers

The MKL optimization problem was first proposed in (Bach et al., 2004) and extended to multiclass in (Zien & Ong, 2007). It can be written as

$$\min_{\bar{\boldsymbol{w}}} \frac{\lambda}{2} \left(\sum_{j=1}^{F} \|\boldsymbol{w}^j\|_2 \right)^2 + \frac{1}{N} \sum_{i=1}^{N} \xi_i \qquad (1)$$
$$\text{s.t. } \bar{\boldsymbol{w}} \cdot (\bar{\boldsymbol{\phi}}(\boldsymbol{x}_i, y_i) - \bar{\boldsymbol{\phi}}(\boldsymbol{x}_i, y)) \geq 1 - \xi_i, \forall i, y \neq y_i \ .$$

An equivalent formulation can be derived from the this one through a variational argument. It has been used in (Bach et al., 2004; Sonnenburg et al., 2006; Rakotomamonjy et al., 2008). The variational formulation allows to use an alternating optimization strategy to efficiently solve the constrained minimization problem. Recently in (Orabona et al., 2010; Jie et al., 2010; Martins et al., 2011) it has been shown that it is possible to efficiently minimize directly the formulation in (1), or at least one variation of it.

We first rewrite (1) with group-norms. Using the notation defined above, we have

$$\min_{\bar{\boldsymbol{w}}} \ \frac{\lambda}{2} \|\bar{\boldsymbol{w}}\|_{2,1}^2 + \frac{1}{N} \sum_{i=1}^{N} \ell^{MC} \left(\bar{\boldsymbol{w}}, \bar{\boldsymbol{\phi}}(\boldsymbol{x}_i, \cdot), y_i \right), \qquad (2)$$

where ℓ^{MC} is the multiclass hinge loss (Crammer & Singer, 2002). The $(2, 1)$ group norm is used to induce sparsity in the domain of the kernels. This means that the solution of the optimization problem will select a subset of the F kernels. However, even if sparsity can be desirable for specific applications, it could bring to a decrease in performance (Kloft et al., 2009; Orabona et al., 2010). Moreover the problem in (2) is not strongly convex (Kakade et al., 2009), so its optimization algorithm is rather complex and its rate of convergence is usually slow (Bach et al., 2004; Sonnenburg et al., 2006). The $(2, p)$ group norm has been proposed instead of the $(2, 1)$ (Kloft et al., 2009; Orabona et al., 2010; Vishwanathan et al., 2010), to be able to decide the level of sparsity of the solution, but this formulation never induces coefficients that are mathematically zero for any $p \neq 1$. It is also interesting to note that the convergence rate of the l_p MKL becomes slower when $p \to 1$ (Orabona et al., 2010). This can be explained formally with the fact that the l_p formulation is $1/q$ strongly convex, where $1/p + 1/q = 1$, and strong convex functions are easier to be optimized. In fact there are optimization algorithms that have a convergence rate proportional to the inverse of the strong convexity constant (Hazan et al., 2007). When p tends to 1, q goes to infinity and the strong convexity is lost, resulting in a slower convergence. In other words, it is more difficult and slower to find a sparse solution to the MKL problem.

Tomioka & Suzuki (2010) have proposed to use an elastic net form of regularization for MKL, that can be written as $C(\frac{\lambda}{2}\|\bar{\boldsymbol{w}}\|_{2,2}^2 + (1-\lambda)\|\bar{\boldsymbol{w}}\|_{2,1})$. They have justified this form of regularization as a mean to control the degree of sparsity of the solution. In this way the solution has exact mathematical zeros, and the number of zeros can be tuned by changing λ.

In the next Section we will introduce the new regularization function, its theoretical properties and its corresponding optimization algorithm.

4. UFO-MKL

Considering the regularizer in (Tomioka & Suzuki, 2010), note that there is no particular reason to use the $(2, 2)$ group norm, apart from having an easier optimization problem and a way to tune the level of sparsity. Similar considerations hold for the $(2, p)$ group norm. Hence, we propose to use a novel regularizer, with the precise aim of having the optimal convergence rate and an exact mathematical sparsity, tunable through a parameter. Our regularization function is

$$\Omega(\bar{\boldsymbol{w}}) := \lambda/2 \ \|\bar{\boldsymbol{w}}\|_{2, \frac{2\log F}{2\log F - 1}}^2 + \alpha \|\bar{\boldsymbol{w}}\|_{2,1}, \qquad (3)$$

where F is the number of kernels. The first term of Ω gives us an easy problem, while the second one induces different levels of sparsity depending on α. In fact, with the l_p MKL formulation, it is possible to prove a convergence bound of the order of

$qF^{2/q}$ (Orabona et al., 2010). If $F \geq 3$ and $p = \frac{2\log F}{2\log F - 1}$, $qF^{2/q}$ becomes equal to $2e\log F$, and the rate of convergence will depend logarithmically on the number of kernels (Jie et al., 2010). More in details, with this choice of p the regularization becomes similar to the entropic regularization. A similar method has been used in the context of sparse linear optimization in (Shalev-Shwartz & Tewari, 2009). This motivates the choice of the first term in Ω. On the other hand, using only this term would result in a fixed regularization function, losing the possibility to adapt it to the problem. Hence we mix the $(2, 2\log F/(2\log F - 1))$ squared group norm with a $(2,1)$ group norm, to be able to tune the level of sparsity.

We propose to use a stochastic gradient descent algorithm, so we can consider a generic loss function ℓ. Hence the optimization problem becomes

$$\min_{\bar{\boldsymbol{w}}} \quad \Omega(\bar{\boldsymbol{w}}) + \frac{1}{N}\sum_{i=1}^{N} \ell\left(\bar{\boldsymbol{w}}, \bar{\boldsymbol{\phi}}(\boldsymbol{x}_i, \cdot), y_i\right) \ . \tag{4}$$

Having designed the regularizer as being strongly convex for any value of α, to minimize (4) we can use stochastic gradient descent and mirror descent. Figure 1 shows the minimization algorithm, which we call Ultra Fast Online Multi Kernel Learning (UFO-MKL) algorithm. As in the mirror descent algorithm, two sets of weights are maintained, a primal one $\bar{\boldsymbol{w}}_t$ and a dual one $\bar{\boldsymbol{\theta}}_t$. At each step it takes a sample at random from the training set and update the dual vector $\bar{\boldsymbol{\theta}}_t$ with a subgradient descent step, where $\partial\ell\left(\bar{\boldsymbol{w}}_t, \bar{\boldsymbol{\phi}}(\boldsymbol{x}_t, \cdot), y_t\right)$ is the subgradient w.r.t. $\bar{\boldsymbol{w}}_t$. Then the primal weight $\bar{\boldsymbol{w}}_t$ is calculated with lines 6-7. These two lines correspond to the gradient of the Fenchel dual of Ω. Line 6 have the effect to put to zeros the kernels that have a norm smaller than αt. It has the effect of inducing exact sparsity in the domain of the kernels. Note that in the algorithm we only need to access to scalar products, so that the kernels can be used without any problem. Even the l_2 norms of $\boldsymbol{\theta}^j_{t+1}$ can be calculated in an efficient incremental way as

$$\|\boldsymbol{\theta}^j_{t+1}\|_2^2 = \|\boldsymbol{\theta}^j_t\|_2^2 - 2\boldsymbol{\theta}^j_t \cdot \boldsymbol{z}^j_t + \|\boldsymbol{z}^j_t\|_2^2 \ .$$

where $\bar{\boldsymbol{z}}_t = \partial\ell\left(\bar{\boldsymbol{w}}_t, \bar{\boldsymbol{\phi}}(\boldsymbol{x}_t, \cdot), y_t\right)$.

4.1. Convergence rate guarantee

In this section we prove a theoretical guarantee for the convergence rate of UFO-MKL to the optimal solution of (4). We use the primal-dual framework for the minimization of regularized loss functions in (Shalev-Shwartz & Kakade, 2008). Note that a similar method has been rediscovered by Xiao (2010). We use Theorem 2 in (Shalev-Shwartz & Kakade, 2008), that for completeness we restate here in a simpler form and with our notation.

Algorithm 1 The UFO-MKL algorithm.

1: **Input:** α, λ, T
2: **Initialize:** $\bar{\boldsymbol{w}}_1 = \boldsymbol{0}$, $\bar{\boldsymbol{\theta}}_1 = \boldsymbol{0}$, $q = 2\log F$
3: **for** $t = 1, 2, \ldots, T$ **do**
4: Sample at random $(\boldsymbol{x}_t, y_t)$
5: $\bar{\boldsymbol{\theta}}_{t+1} = \bar{\boldsymbol{\theta}}_t - \partial\ell\left(\bar{\boldsymbol{w}}_t, \bar{\boldsymbol{\phi}}(\boldsymbol{x}_t, \cdot), y_t\right)$
6: $v_j = \left|\|\boldsymbol{\theta}^j_{t+1}\|_2 - \alpha t\right|_+, \forall j = 1, \cdots, F$
7: $\boldsymbol{w}^j_{t+1} = \frac{v_j \boldsymbol{\theta}^j_{t+1}}{t\lambda\|\boldsymbol{\theta}^j_{t+1}\|_2}\left(\frac{v_j}{\|\boldsymbol{v}\|_q}\right)^{q-2}, \ \forall j = 1, \cdots, F$
8: **end for**

Theorem 1. *(Shalev-Shwartz & Kakade, 2008) Let g be a β-strongly convex function w.r.t. the norm $\|\cdot\|$ over a set S and let $\|\cdot\|_*$ be its dual norm. Let $\ell_1, \ldots, \ell_T$ be an arbitrary sequence of convex loss functions, and R such that $\max_i \|\partial\ell_i(\boldsymbol{w}_i)\|_* \leq R$. Define $\boldsymbol{w}_t = \nabla g^*(-\frac{\eta}{t}\sum_{i=1}^{t-1}\partial\ell_i(\boldsymbol{w}_i))$ then, for any $\boldsymbol{u} \in S$, and any $\eta > 0$ we have*

$$\frac{1}{T}\sum_{t=1}^{T}\left(\frac{g(\boldsymbol{w}_t)}{\eta} + \ell_t(\boldsymbol{w}_t)\right) - \frac{1}{T}\sum_{t=1}^{T}\left(\frac{g(\boldsymbol{u})}{\eta} + \ell_t(\boldsymbol{u})\right) \leq \eta\frac{R^2(1+\log T)}{2\beta T} \ .$$

This theorem introduces another way to minimize strongly convex regularized objective functions through stochastic gradient descent, different from the well-known one used in the Pegasos algorithm (Shalev-Shwartz et al., 2007). Here there is no rescaling of the hyperplane at each step, so each single iteration will be faster. This lack of rescaling makes each iteration of UFO-MKL faster than each iteration of OBSCURE (Orabona et al., 2010), so that the total time needed to converge can be smaller. We will verify this experimentally in Section 5.3.

Denote by $\bar{\boldsymbol{z}} = \partial\ell\left(\bar{\boldsymbol{w}}, \bar{\boldsymbol{\phi}}(\boldsymbol{x}, \cdot), y\right)$, we will now state the convergence theorem for any loss function that satisfies the following hypothesis

$$\|\boldsymbol{z}^j\|_2 \leq L\|\boldsymbol{\phi}^j(\boldsymbol{x}, y')\|_2, \ \forall j = 1, \ldots, F, \ y' \in \mathbb{Y} \ . \tag{5}$$

Note that, for example, the hinge loss, $\ell^{HL}\left(\bar{\boldsymbol{w}}, \bar{\boldsymbol{\phi}}(\boldsymbol{x}), y\right) := |1 - y\bar{\boldsymbol{w}} \cdot \bar{\boldsymbol{\phi}}(\boldsymbol{x})|_+$, and the logistic loss, $\ell^{LL}\left(\bar{\boldsymbol{w}}, \bar{\boldsymbol{\phi}}(\boldsymbol{x}), y\right) := \log(1 + \exp(-y\bar{\boldsymbol{w}} \cdot \bar{\boldsymbol{\phi}}(\boldsymbol{x})))$, satisfy this relation with $L = 1$. The multiclass hinge loss function (Crammer & Singer, 2002; Tsochantaridis et al., 2004), $\ell^{MC}\left(\bar{\boldsymbol{w}}, \bar{\boldsymbol{\phi}}(\boldsymbol{x}, \cdot), y\right) := \max_{y' \neq y}|1 - \bar{\boldsymbol{w}} \cdot (\bar{\boldsymbol{\phi}}(\boldsymbol{x}, y) - \bar{\boldsymbol{\phi}}(\boldsymbol{x}, y'))|_+$, satisfies with $L = \sqrt{2}$ when $\boldsymbol{\phi}(\cdot, \cdot)$ induces the transformation in which there is one hyperplane for each class.

Theorem 2. *Denote by* $f(\bar{\boldsymbol{w}}) = \Omega(\bar{\boldsymbol{w}}) + \frac{1}{N}\sum_{i=1}^{N} \ell\left(\bar{\boldsymbol{w}}, \bar{\boldsymbol{\phi}}(\boldsymbol{x}_i, \cdot), y_i\right)$ *and by* $\bar{\boldsymbol{w}}^*$ *the solution that minimizes (4). Suppose that* $\|\boldsymbol{\phi}^j(\boldsymbol{x}_t, \cdot)\|_2 \leq 1$, *and the loss function* ℓ *satisfies (5). Let* $\delta \in (0,1)$, *then with probability at least* $1-\delta$ *over the choices of the random samples we have that after* T *iterations of the UFO-MKL algorithm*

$$f(\bar{\boldsymbol{w}}_{T+1}) - f(\bar{\boldsymbol{w}}^*) \leq \frac{eL^2(1+\log T)\log F}{\lambda \delta T},$$

where e *is the Euler's number.*

Proof. (Sketch) Using (5) and $\|\boldsymbol{\phi}^j(\boldsymbol{x}_t, y_t)\|_2 \leq 1$, we have

$$\begin{aligned}&\|\partial \ell\left(\bar{\boldsymbol{w}}_t, \bar{\boldsymbol{\phi}}(\boldsymbol{x}_t, \cdot), y_t\right)\|_{2,q} \\ &\leq LF^{1/q} \max_{j=1,\ldots,F} \|\boldsymbol{\phi}^j(\boldsymbol{x}_t, \cdot)\|_2 \leq LF^{1/q}\end{aligned}$$

The function $\Omega(\bar{\boldsymbol{w}})$ in (3) is λ/q-strongly convex w.r.t. the norm $\|\cdot\|_{2,q}$, for any $\alpha \geq 0$. Hence, using Theorem 1, with $\eta = 1$ and $g = \Omega$, and using Markov inequality as in (Shalev-Shwartz et al., 2007) we prove the stated result. □

To derive the UFO-MKL algorithm the only thing that is missing is to calculate $\nabla\Omega^*(\bar{\boldsymbol{\theta}})$.

Theorem 3. *Let*

$$\boldsymbol{v} = \left[\left|\|\boldsymbol{\theta}^1\|_2 - \alpha\right|_+, \cdots, \left|\|\boldsymbol{\theta}^F\|_2 - \alpha\right|_+\right],$$

then the component j *of* $\nabla\Omega^*(\bar{\boldsymbol{\theta}})$ *is equal to*

$$\frac{\boldsymbol{\theta}^j}{\lambda\|\boldsymbol{\theta}^j\|_2} \frac{v_j^{q-1}}{\|\boldsymbol{v}\|_q^{q-2}}$$

Proof. (Sketch) From standard Legendre-Fenchel duality, we have that $\nabla\Omega^*(\bar{\boldsymbol{\theta}}) = \operatorname{argmax}_{\bar{\boldsymbol{w}}} \bar{\boldsymbol{w}} \cdot \bar{\boldsymbol{\theta}} - \Omega(\bar{\boldsymbol{w}})$. Setting to zero the derivative of this argmax we have that $\boldsymbol{w}^j$ must must be proportional to $\boldsymbol{\theta}^j$, that is $\boldsymbol{w}^j = c_j\boldsymbol{\theta}^j/\|\boldsymbol{\theta}^j\|$, where c_j are real numbers. So we can focus on the coefficients c_j, rewriting the argmax:

$$\operatorname*{argmax}_{\boldsymbol{c}} \boldsymbol{c}\cdot\boldsymbol{a} - \alpha\|\boldsymbol{c}\|_1 - \lambda/2\ \|\boldsymbol{c}\|_p^2,$$

where $\boldsymbol{a} = [\|\boldsymbol{\theta}_1\|, \cdots, \|\boldsymbol{\theta}_F\|]$, $\boldsymbol{c} = [c_1, \cdots, c_F]$. This problem is analyzed in Sec. 7.2 of (Xiao, 2010), and using that theorems we have the stated result. □

5. Experiments

In this section, we study the behavior of UFO-MKL in terms of classification accuracy, computational efficiency and scalability. Our algorithm has been implemented in MATLAB in the DOGMA library (Orabona, 2009), and we compare it against the SHOGUN-0.9.2 toolbox[2], which contains the SILP algorithm (Sonnenburg et al., 2006) and the Multiclass MKL (MC-MKL) algorithm (Zien & Ong, 2007). We also compare it with the OBSCURE algorithm (Orabona et al., 2010), using the implementation in DOGMA. We consider the parameter $\alpha \in \{0.0001, 0.001, 0.0025, 0.005, 0.0075, 0.01, 0.02\}$, and use p=1.01 for the OBSCURE algorithm when sparsity is desired in the solution. The λ parameter has been chosen by cross validation as $1/(CN)$, where N is the number of training points, and C is from the set $\{1, 10, 10^2, 10^3\}$, and C=1000 yields the best results for all the algorithms, except in the first and second experiment where we fix $C = 100$.

We consider both hinge loss ℓ^{HL} and logistical loss ℓ^{LL} for the binary classification task, and multiclass loss ℓ^{MC} for all the multiple classes tasks. For multiple classes extension of the binary SILP algorithm, we use the 1-vs-All scheme.

5.1. Binary classification

We first carry out a set of experiments on the UCI binary data sets, and compare the results with SILP. We follow the procedure in (Rakotomamonjy et al., 2008), to test the algorithms with artificially generated kernels. The candidate kernels are Gaussian kernels with 10 different bandwidths on all and each single dimension of the feature vectors, and similar for polynomial kernels of degree 1 to 3. In UFO-MKL we use 10 epochs, that is 10 passes over all the training samples, for the Liver and Ionosphere dataset, and 20 epochs for the Sonar dataset. Results for varying values of α are presented in Table 1. We can see that UFO-MKL is significantly faster compared to SILP when used with the hinge loss, especially on Sonar where it is more than 10 times faster. In most cases SILP gets a sparser solution than UFO-MKL's one, which is due to the different regularizer used. It is known that l_1 norm kind of regularizer can result in bad performance when the problem is not sparse (Kloft et al., 2009; Orabona et al., 2010). However, with proper tuning of α, UFO-MKL can possibly still remove some of the kernels, while still obtain superior performance (see for example $\alpha = 1.0e-3$ on the Sonar dataset).

5.2. Multiclass synthetic data

Multiclass problems are often decomposed into several binary sub-problems using methods like 1-vs-All, how-

[2] Available at `http://www.shogun-toolbox.org`.

Table 1. Training time, accuracy and number of selected kernels on UCI datasets with N samples and F kernels.

Loss	α	Time (seconds)	Accuracy	# Kernels$\neq 0$
Liver, N=241, F=91				
ℓ^{HL}	1.0e−3	2.7 ± 0.3	71.8 ± 3.8	89.7 ± 0.6
	2.5e−3	2.9 ± 0.3	66.1 ± 2.7	30.5 ± 3.5
ℓ^{LL}	1.0e−3	3.5 ± 0.4	68.7 ± 4.1	91.0 ± 0.0
	2.5e−3	3.6 ± 0.4	64.5 ± 3.4	28.1 ± 15.4
SILP		3.0 ± 0.6	64.7 ± 1.8	9.9 ± 1.8
Ionosphere, N=245, F=442				
ℓ^{HL}	1.0e−3	6.1 ± 0.8	92.1 ± 2.1	257.7 ± 87.0
	2.5e−3	6.1 ± 0.3	91.4 ± 1.7	93.2 ± 7.9
	5.0e−3	8.2 ± 0.5	89.2 ± 2.4	37.4 ± 5.6
ℓ^{LL}	1.0e−3	21.9 ± 2.0	91.8 ± 2.6	442 ± 0.0
	2.5e−3	18.2 ± 1.4	87.8 ± 2.0	119.7 ± 6.3
	5.0e−3	19.2 ± 0.9	86.4 ± 3.6	1.6 ± 2.5
SILP		41.9 ± 9.5	91.9 ± 2.1	19.6 ± 3.0
Sonar, N=145, F=793				
ℓ^{HL}	1.0e−3	9.9 ± 0.5	80.3 ± 3.5	379.7 ± 131.1
	2.5e−3	10.9 ± 0.9	79.6 ± 4.7	192.6 ± 12.5
	5.0e−3	11.7 ± 0.5	75.7 ± 2.4	55.6 ± 9.1
ℓ^{LL}	1.0e−3	22.4 ± 2.1	77.0 ± 4.6	793.0 ± 0.0
	2.5e−3	21.9 ± 1.5	75.5 ± 4.1	200.1 ± 64.8
	5.0e−3	21.9 ± 1.8	74.4 ± 4.5	28.8 ± 4.3
SILP		191.2 ± 38.7	78.7 ± 3.8	30.0 ± 2.2

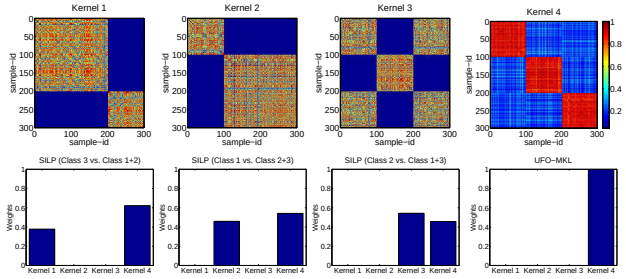

Figure 1. (top) Kernel matrices of the 3-classes synthetic experiments correspond to 4 different features. Sample 1–100, 101–200 and 201–300 are from class 1, 2 and 3 respectively. (bottom) Corresponding kernel combination weights, normalized to have sum equal to 1, obtained by SILP (binary) and by UFO-MKL (multiclass) (last figure).

ever solving the multiclass learning problem jointly using a multiclass loss can yield much sparser solutions. Intuitively, when l_1-norm is used to impose sparsity in the domain of kernels, different subsets of kernels can be selected for different binary classification problems. Therefore, the combined multiclass classifier might not obtain the desired sparse properties. Moreover, the confidence outputs of the binary classifiers may not lie in the same range, so it is not clear if the winner-takes-all hypothesis is the correct approach for combing them.

To prove our point, we have generated a 3-classes classification problem consisting of 300 samples, with 100 sample from each class. There are in total 4 different features, the kernel matrices corresponding to them are shown in Figure 1 (top). These features are generated in a way that Kernels 1–3 are useful only for

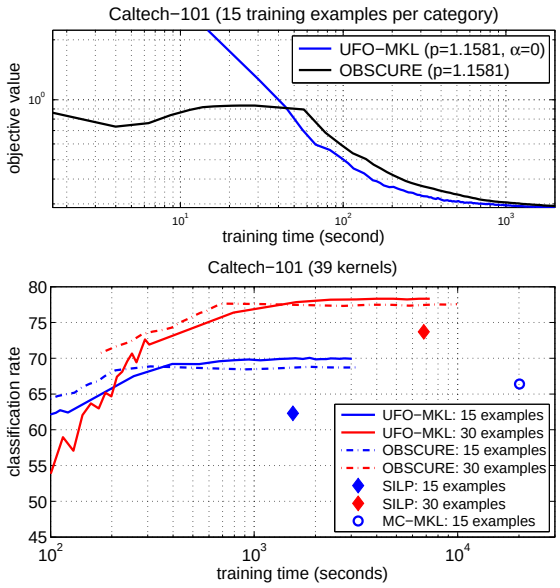

Figure 2. (Top) Comparison of UFO-MKL to OBSCURE on convergence rate with the same value of p, and $\alpha = 0$ for UFO-MKL. (Bottom) Performance comparison on Caltech-101 using different MKL algorithms.

distinguishing one class (class 3, class 1 and class 2, respectively) from the other two, while Kernel 4 can separate all the 3 classes. The corresponding kernel combination weights obtained by the SILP algorithm using the 1-vs-All extension and our multiclass UFO-MKL are shown in Figure 1 (bottom). It can be observed that each of the binary SILP classifiers pick two kernels. UFO-MKL selects only the 4th kernel, achieving a much sparser solution.

5.3. Multiple classes image categorization

Most of the state-of-art results obtained on several object categorization datasets use algorithms which combine multiple features (see (Gehler & Nowozin, 2009; Orabona et al., 2010) and references therein). The Caltech-101 dataset is the most popular benchmark for object categorization. In the experiments, we used the pre-computed features and kernels of (Gehler & Nowozin, 2009) which the authors have made available[3]. There are in total 39 kernels, with 5 different training and test splits. The best results on this data were obtained using regularization which favors sparsity (Gehler & Nowozin, 2009; Orabona et al., 2010). For brevity, we refer the interested reader to (Gehler & Nowozin, 2009) for the details of the features and kernels.

[3] www.vision.ee.ethz.ch/~pgehler/projects/iccv09/

We start by comparing the convergence rates of UFO-MKL and OBSCURE, which is the state-of-art p-norm multiclass MKL solver. The training time of the OBSCURE algorithm is proportional to q/λ, where $1/p + 1/q = 1$. Therefore, when a sparse solution is needed, the algorithm becomes slow because q becomes big. For a fair comparison, we first set $q = 2 \log F$ in OBSCURE, and $\alpha = 0$ in UFO-MKL, so that their regularizers become exactly the same. Figure 2 (top) shows the value of the objective function as a function of the training time. OBSCURE is faster in the beginning because its first stage is an online algorithm, which quickly determines the region of the space where the optimal solution lives. UFO-MKL, after $\approx$ 1min of computation, converges faster than OBSCURE. We think that this is due to the simpler algorithm that does not require a scaling after each update, hence each iteration in UFO-MKL is faster.

Following the experimental setup widely used in the computer vision literatures, we also report the results obtained using different MKL methods on various number of training data in Figure 2 (bottom). The results support our claim of the previous section that multiclass loss function is more suitable for this type of problem, as all the methods that use the multiclass loss outperform SILP. MC-MKL is computational infeasible for 30 samples per category, and its significant performance gap from OBSCURE and UFO-MKL seems to indicate that it stops before converging to the optimal solution. UFO-MKL also outperforms OBSCURE, probably because OBSCURE does not get a real sparse solution although it tends to be sparse. More importantly, the accuracy is comparable with the best results obtained in the literature by the LP-β algorithm (Gehler & Nowozin, 2009) using the same kernels: 70.4% for 15 training samples per category and 77.8% for 30 samples.

5.4. Scalability w.r.t. the number of kernels

To test the scalability of UFO-MKL we tested it on the Oxford Flower data set (Nilsback & Zisserman, 2006)[4], generating 1400 kernels. The task of the dataset is to classify 17 different flower categories. Each class has 80 images with predefined train and test splits. Precomputed distance matrices for 7 different features are available. For each precomputed matrix, we generates 200 kernels using $\exp(-\gamma^{-1} \cdot d)$ with 200 different γ values in the range between 0.01 and 100. Figure 3 (top) reports the results for varying number of kernels. Our algorithm outperforms all the other baseline in term of both accuracy and efficiency.

[4] www.robots.ox.ac.uk/~vgg/research/flowers/

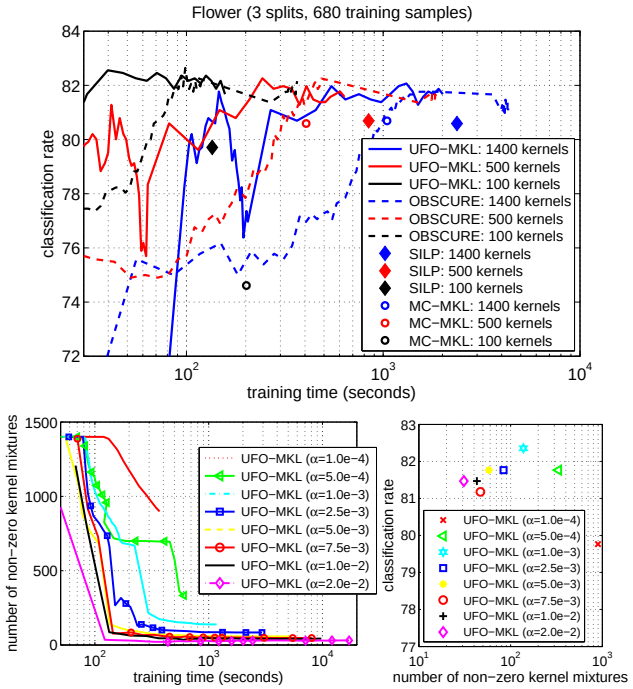

Figure 3. Running time performance of UFO-MKL and other baseline approaches w.r.t. different number of kernels and varying values of α.

UFO-MKL is 3-5 times faster compared to OBSCURE, which is again due to the factor that OBSCURE does not get a real sparse solution. It suggests that UFO-MKL is more suitable for feature selection tasks when a lot of kernels are available. Figure 3 (bottom left) shows the number of selected kernels and the accuracy obtained by varying values of α, using the same number of epochs. We see that a larger value of α, which corresponds to a sparser solution, leads to a slower running time. Figure 3 (bottom right) reports the accuracy and the number of non-zero kernel weights of the last solution when the algorithm stops. It can be seen that when the model becomes over sparse (larger α) the performance starts dropping. However small values of α, which result in a denser model, do not correspond to higher accuracy, in fact many less discriminative kernels are included in the solution.

6. Conclusions and Discussion

This paper presents a new MKL formulation and a fast algorithm, UFO-MKL, to solve it. It optimizes the objective function directly in the primal with a stochastic subgradient descent method. Experiments show that UFO-MKL achieves state-of-art performance on binary and multiclass classification problems. Our approach is general, hence it can be applied to any

convex loss function such as *structure output prediction* (Tsochantaridis et al., 2004), to have an MKL algorithm for structured output.

UFO-MKL has a guaranteed convergence rate, with a logarithmic dependence on the number of kernels. Moreover the level of sparsity is tunable, and the solution found will always have exact zeros.

Acknowledgments. FO and LJ acknowledge support by the Pascal Pump Priming Project SS2-Rob funded by PASCAL2 NoE under EC grant FP7-216886.

References

Bach, F. R., Lanckriet, G. R. G., and Jordan, M. I. Multiple kernel learning, conic duality, and the SMO, algorithm. In *Proc. of the 21th Intl Conf. on Machine Learning*, 2004.

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

Chapelle, O. Training a support vector machine in the primal. *Neural Comput.*, 19:1155–1178, 2007.

Cortes, C., Mohri, M., and Rostamizadeh, A. Two-stage learning kernel algorithms. In *Proc. of the 27th Intl Conf. on Machine Learning*, 2010.

Crammer, K. and Singer, Y. On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res.*, 2:265–292, 2002.

Cristianini, N. and Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.

Cristianini, N., Kandola, J., Elisseeff, A., and Shawe-Taylor, J. On kernel-target alignment. In *Adv. Neural. Inform. Process Syst. 14*, 2002.

Gehler, P. and Nowozin, S. On feature combination for multiclass object classification. In *Proc. of IEEE Intl Conf. on Computer Vision*, 2009.

Hazan, E., Agarwal, A., and Kale, S. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

Jie, L., Orabona, F., Fornoni, M., Caputo, B., and Cesa-Bianchi, N. OM-2: An online multi-class multi-kernel learning algorithm. In *Proc. of the 4th IEEE Online Learning for Computer Vision Workshop (in CVPR 2010)*, 2010.

Kakade, S., Shalev-Shwartz, S., and Tewari, A. On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization. Technical report, TTI, 2009.

Kloft, M., Brefeld, U., Sonnenburg, S., Laskov, P., Müller, K.-R., and Zien, A. Efficient and accurate lp-norm multiple kernel learning. In *Adv. Neural. Inform. Process Syst. 22*. 2009.

Lanckriet, G., Cristianini, N., Bartlett, P., and Ghaoui, L. E. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5, 2004.

Martins, A. F. T., Smith, N. A., Xing, E. P., Aguiar, P., and Figueiredo, M. Online learning of structured predictors with multiple kernels. In *Proc. of the 14th Intl Conf. on Artificial Intelligence and Statistics*, 2011.

Nath, J. S., Dinesh, G., Raman, S., Bhattacharyya, C., Ben-Tal, A., and Ramakrishnan, K. R. On the algorithmics and applications of a mixed-norm based kernel learning formulation. In *Adv. Neural. Inform. Process Syst. 22*, 2009.

Nilsback, M.-E. and Zisserman, A. A visual vocabulary for flower classification. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

Orabona, F. *DOGMA: a MATLAB Toolbox for Online Learning*, 2009. Software available at `http://dogma.sourceforge.net`.

Orabona, F., Jie, L., and Caputo, B. Online-batch strongly convex multi kernel learning. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.

Rakotomamonjy, A., Bach, F. R., Canu, S., and Grandvalet, Y. SimpleMKL,. *J. Mach. Learn. Res.*, 9:2491–2521, 2008.

Shalev-Shwartz, S. and Kakade, S. M. Mind the duality gap: Logarithmic regret algorithms for online optimization. In *Adv. Neural. Inform. Process Syst. 21*, 2008.

Shalev-Shwartz, S. and Tewari, A. Stochastic methods for l1 regularized loss minimization. In *Proc. of the 26th Intl Conf. on Machine Learning*, 2009.

Shalev-Shwartz, S., Singer, Y., and Srebro, N. Pegasos:Primal Estimated sub-GrAdient SOlver for SVM. In *Proc. of the 24th Intl Conf. on Machine Learning*, 2007.

Sonnenburg, S., Rätsch, G., Schäfer, C., and Schölkopf, B. Large scale multiple kernel learning. *J. Mach. Learn. Res.*, 7:1531–1565, 2006.

Tomioka, R. and Suzuki, T. Sparsity-accuracy trade-off in MKL, 2010. URL `http://arxiv.org/abs/1001.2615`.

Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. Support vector machine learning for interdependent and structured output spaces. In *Proc. of the 21th Intl Conf. on Machine Learning*, 2004.

Vishwanathan, S. V. N., Sun, Z., Theera-Ampornpunt, N., and Varma, M. Multiple kernel learning and the SMO algorithm. In *Adv. Neural. Inform. Process Syst.*, 2010.

Xiao, L. Dual averaging methods for regularized stochastic learning and online optimization. *J. Mach. Learn. Res.*, 11:2543–2596, October 2010.

Xu, Z., Jin, R., King, I., and Lyu, M. R. An extended level method for efficient multiple kernel learning. In *Adv. Neural. Inform. Process Syst. 21*, 2008.

Zien, A. and Ong, C. S. Multiclass multiple kernel learning. In *Proc. of the 24th Intl Conf. on Machine Learning*, 2007.