Efficient and Accurate ℓ_p-Norm Multiple Kernel Learning

Marius Kloft
University of California
Berkeley, USA

Ulf Brefeld
Yahoo! Research
Barcelona, Spain

Sören Sonnenburg
Technische Universität Berlin
Berlin, Germany

Pavel Laskov
Universität Tübingen
Tübingen, Germany

Klaus-Robert Müller
Technische Universität Berlin
Berlin, Germany

Alexander Zien
LIFE Biosystems GmbH
Heidelberg, Germany

Abstract

Learning linear combinations of multiple kernels is an appealing strategy when the right choice of features is unknown. Previous approaches to multiple kernel learning (MKL) promote sparse kernel combinations to support interpretability. Unfortunately, ℓ_1-norm MKL is hardly observed to outperform trivial baselines in practical applications. To allow for robust kernel mixtures, we generalize MKL to arbitrary ℓ_p-norms. We devise new insights on the connection between several existing MKL formulations and develop two efficient *interleaved* optimization strategies for arbitrary $p > 1$. Empirically, we demonstrate that the interleaved optimization strategies are much faster compared to the traditionally used wrapper approaches. Finally, we apply ℓ_p-norm MKL to real-world problems from computational biology, showing that non-sparse MKL achieves accuracies that go beyond the state-of-the-art.

1 Introduction

Sparseness is being regarded as one of the key features in machine learning [15] and biology [16]. Sparse models are appealing since they provide an intuitive interpretation of a task at hand by singling out relevant pieces of information. Such automatic complexity reduction facilitates efficient training algorithms, and the resulting models are distinguished by small capacity. The interpretability is one of the main reasons for the popularity of sparse methods in complex domains such as computational biology, and consequently building sparse models from data has received a significant amount of recent attention.

Unfortunately, sparse models do not always perform well in practice [7, 15]. This holds particularly for learning sparse linear combinations of data sources [15], an abstraction of which is known as multiple kernel learning (MKL) [10]. The data sources give rise to a set of (possibly correlated) kernel matrices $K_1, \ldots, K_M$, and the task is to learn the optimal mixture $K = \sum_m \theta_m K_m$ for the problem at hand. Previous MKL research aims at finding sparse mixtures to effectively simplify the underlying data representation. For instance, [10] study semi-definite matrices $K \succeq 0$ inducing sparseness by bounding the trace $\mathrm{tr}(K) \leq c$; unfortunately, the resulting semi-definite optimization problems are computationally too expensive for large-scale deployment.

Recent approaches to MKL promote sparse solutions either by Tikhonov regularization over the mixing coefficients [25] or by incorporating an additional constraint $\|\boldsymbol{\theta}\| \leq 1$ [18, 27] requiring solutions on the standard simplex, known as Ivanov regularization. Based on the one or the other, efficient optimization strategies have been proposed for solving ℓ_1-norm MKL using semi-infinite linear programming [21], second order approaches [6], gradient-based optimization [19], and level-set methods [26]. Other variants of ℓ_1-norm MKL have been proposed in subsequent work addressing practical algorithms for multi-class [18, 27] and multi-label [9] problems.

Previous approaches to MKL successfully identify sparse kernel mixtures, however, the solutions found, frequently suffer from poor generalization performances. Often, trivial baselines using unweighted-sum kernels $K = \sum_m K_m$ are observed to outperform the sparse mixture [7]. One reason for the collapse of ℓ_1-norm MKL is that kernels deployed in real-world tasks are usually highly sophisticated and effectively capture relevant aspects of the data. In contrast, sparse approaches to MKL rely on the assumption that some kernels are irrelevant for solving the problem. Enforcing sparse mixtures in these situations may lead to degenerate models. As a remedy, we propose to sacrifice sparseness in these situations and deploy non-sparse mixtures instead. After submission of this paper, we learned about a related approach, in which the sum of an ℓ_1- and an ℓ_2-regularizer are used [12]. Although non-sparse solutions are not as easy to interpret, they account for (even small) contributions of all available kernels to live up to practical applications.

In this paper, we first show the equivalence of the most common approaches to ℓ_1-norm MKL [18, 25, 27]. Our theorem allows for a generalized view of recent strands of multiple kernel learning research. Based on the detached view, we extend the MKL framework to arbitrary ℓ_p-norm MKL with $p \geq 1$. Our approach can either be motivated by additionally regularizing over the mixing coefficients $\|\boldsymbol{\theta}\|_p^p$, or equivalently by incorporating the constraint $\|\boldsymbol{\theta}\|_p^p \leq 1$. We propose two alternative optimization strategies based on Newton descent and cutting planes, respectively. Empirically, we demonstrate the efficiency and accuracy of none-sparse MKL. Large-scale experiments on gene start detection show a significant improvement of predictive accuracy compared to ℓ_1- and ℓ_∞-norm MKL.

The rest of the paper is structured as follows. We present our main contributions in Section 2, the theoretical analysis of existing approaches to MKL, our ℓ_p-norm MKL generalization with two highly efficient optimization strategies, and relations to ℓ_1-norm MKL. We report on our empirical results in Section 3 and Section 4 concludes.

2 Generalized Multiple Kernel Learning

2.1 Preliminaries

In the standard supervised learning setup, a labeled sample $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1\ldots,n}$ is given, where the $\boldsymbol{x}$ lie in some input space $\mathcal{X}$ and $y \in \mathcal{Y} \subset \mathbb{R}$. The goal is to find a hypothesis $f \in \mathcal{H}$, that generalizes well on new and unseen data. Applying *regularized risk minimization* returns the minimizer f^*,

$$f^* = \mathrm{argmin}_f \ \mathrm{R}_{\text{emp}}(f) + \lambda\Omega(f),$$

where $\mathrm{R}_{\text{emp}}(f) = \frac{1}{n}\sum_{i=1}^n V\left(f(\boldsymbol{x}_i), y_i\right)$ is the empirical risk of hypothesis f w.r.t. to the loss $V : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$, regularizer $\Omega : \mathcal{H} \to \mathbb{R}$, and trade-off parameter $\lambda > 0$. In this paper, we focus on $\Omega(f) = \frac{1}{2}\|\tilde{\boldsymbol{w}}\|_2^2$ and on linear models of the form

$$f_{\tilde{\boldsymbol{w}},b}(\boldsymbol{x}) = \tilde{\boldsymbol{w}}^\top \psi(\boldsymbol{x}) + b, \tag{1}$$

together with a (possibly non-linear) mapping $\psi : \mathcal{X} \to \mathcal{H}$ to a Hilbert space $\mathcal{H}$ [20]. We will later make use of kernel functions $K(\boldsymbol{x}, \boldsymbol{x}') = \langle\psi(\boldsymbol{x}), \psi(\boldsymbol{x}')\rangle_{\mathcal{H}}$ to compute inner products in $\mathcal{H}$.

2.2 Learning with Multiple Kernels

When learning with multiple kernels, we are given M different feature mappings $\psi_m : \mathcal{X} \to \mathcal{H}_m,\ m = 1, \ldots M$, each giving rise to a reproducing kernel K_m of $\mathcal{H}_m$. Approaches to *multiple kernel learning* consider linear kernel mixtures $K_{\boldsymbol{\theta}} = \sum \theta_m K_m$, $\theta_m \geq 0$. Compared to Eq. (1), the primal model for learning with multiple kernels is extended to

$$f_{\tilde{\boldsymbol{w}},b,\boldsymbol{\theta}}(\boldsymbol{x}) \ = \ \tilde{\boldsymbol{w}}^\top \psi_{\boldsymbol{\theta}}(\boldsymbol{x}_i) + b \ = \ \sum_{m=1}^M \sqrt{\theta_m}\tilde{\boldsymbol{w}}_m^\top \psi_m(\boldsymbol{x}) + b, \tag{2}$$

where the weight vector $\tilde{\boldsymbol{w}}$ and the composite feature map $\psi_{\boldsymbol{\theta}}$ have a block structure $\tilde{\boldsymbol{w}} = (\tilde{\boldsymbol{w}}_1^\top, \ldots, \tilde{\boldsymbol{w}}_M^\top)^\top$ and $\psi_{\boldsymbol{\theta}} = \sqrt{\theta_1}\psi_1 \times \ldots \times \sqrt{\theta_M}\psi_M$, respectively.

The idea in learning with multiple kernels is to minimize the loss on the training data w.r.t. to optimal kernel mixture $\sum \theta_m K_m$ in addition to regularizing $\boldsymbol{\theta}$ to avoid overfitting. Hence, in terms of regularized risk minimization, the optimization problem becomes

$$\inf_{\tilde{\boldsymbol{w}},b,\boldsymbol{\theta}\geq\mathbf{0}} \quad \frac{1}{n}\sum_{i=1}^{n} V\left(f_{\boldsymbol{w},b,\boldsymbol{\theta}}(\boldsymbol{x}_i),\ y_i\right) + \frac{\lambda}{2}\sum_{m=1}^{M} \|\tilde{\boldsymbol{w}}_m\|_2^2 + \tilde{\mu}\tilde{\Omega}[\boldsymbol{\theta}]. \tag{3}$$

Previous approaches to multiple kernel learning employ regularizers of the form $\tilde{\Omega}(\boldsymbol{\theta}) = ||\boldsymbol{\theta}||_1$ to promote sparse kernel mixtures. By contrast, we propose to use smooth convex regularizers of the form $\tilde{\Omega}(\boldsymbol{\theta}) = ||\boldsymbol{\theta}||_p^p$, $1 < p < \infty$, allowing for non-sparse solutions. The non-convexity of the resulting optimization problem is not inherent and can be resolved by substituting $\boldsymbol{w}_m \leftarrow \sqrt{\theta_m}\tilde{\boldsymbol{w}}_m$. Furthermore, regularization parameter and sample size can be decoupled by introducing $\tilde{C} = \frac{1}{n\lambda}$ (and adjusting $\mu \leftarrow \frac{\tilde{\mu}}{\lambda}$) which has favorable scaling properties in practice. We obtain the following convex optimization problem [5] that has also been considered by [25] for hinge loss and $p = 1$,

$$\inf_{\boldsymbol{w},b,\boldsymbol{\theta}\geq\mathbf{0}} \quad \tilde{C}\sum_{i=1}^{n} V\left(\sum_{m=1}^{M} \boldsymbol{w}_m^\top \psi_m(\boldsymbol{x}_i) + b,\ y_i\right) + \frac{1}{2}\sum_{m=1}^{M} \frac{\|\boldsymbol{w}_m\|_2^2}{\theta_m} + \mu||\boldsymbol{\theta}||_p^p, \tag{4}$$

where we use the convention that $\frac{t}{0} = 0$ if $t = 0$ and ∞ otherwise. An alternative approach has been studied by [18, 27] (again using hinge loss and $p = 1$). They upper bound the value of the regularizer $\|\boldsymbol{\theta}\|_1 \leq 1$ and incorporate the latter as an additional constraint into the optimization problem. For $C > 0$, they arrive at

$$\inf_{\boldsymbol{w},b,\boldsymbol{\theta}\geq\mathbf{0}} \quad C\sum_{i=1}^{n} V\left(\sum_{m=1}^{M} \boldsymbol{w}_m^\top \psi_m(\boldsymbol{x}_i) + b,\ y_i\right) + \frac{1}{2}\sum_{m=1}^{M} \frac{||\boldsymbol{w}_m||_2^2}{\theta_m} \qquad \text{s.t.} \quad ||\boldsymbol{\theta}||_p^p \leq 1. \tag{5}$$

Our first contribution shows that both, the Tikhonov regularization in Eq. (4) and the Ivanov regularization in Eq. (5), are equivalent.

Theorem 1 *Let be $p \geq 1$. For each pair $(\tilde{C}, \mu)$ there exists $C > 0$ such that for each optimal solution $(\boldsymbol{w}^*, b^*, \boldsymbol{\theta}^*)$ of Eq. (4) using $(\tilde{C}, \mu)$, we have that $(\boldsymbol{w}^*, b^*, \kappa\,\boldsymbol{\theta}^*)$ is also an optimal solution of Eq. (5) using C, and vice versa, where $\kappa > 0$ is some multiplicative constant.*

Proof. The proof is shown in the supplementary material for lack of space. Sketch of the proof: We incorporate the regularizer of (4) into the constraints and show that the resulting upper bound is tight. A variable substitution completes the proof. □

Zien and Ong [27] showed that the MKL optimization problems by Bach et al. [3], Sonnenburg et al. [21], and their own formulation are equivalent. As a main implication of Theorem 1 and by using the result of Zien and Ong it follows that the optimization problem of Varma and Ray [25] and the ones from [3, 18, 21, 27] all are equivalent.

In addition, our result shows the coupling between trade-off parameter C and the regularization parameter μ in Eq. (4): tweaking one also changes the other and vice versa. Moreover, Theorem 1 implies that optimizing C in Eq. (5) implicitly searches the regularization path for the parameter μ of Eq. (4). In the remainder, we will therefore focus on the formulation in Eq. (5), as a single parameter is preferable in terms of model selection. Furthermore, we will focus on binary classification problems with $\mathcal{Y} = \{-1, +1\}$, equipped with the hinge loss $V(f(\boldsymbol{x}), y) = \max\{0, 1 - yf(\boldsymbol{x})\}$. However note, that all our results can easily be transferred to regression and multi-class settings using appropriate convex loss functions and joint kernel extensions.

2.3 Non-Sparse Multiple Kernel Learning

We now extend the existing MKL framework to allow for non-sparse kernel mixtures $\boldsymbol{\theta}$, see also [13]. Let us begin with rewriting Eq. (5) by expanding the hinge loss into the slack variables as follows

$$\min_{\boldsymbol{\theta},\boldsymbol{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\sum_{m=1}^{M} \frac{||\boldsymbol{w}_m||_2^2}{\theta_m} + C\|\boldsymbol{\xi}\|_1 \tag{6}$$

$$\text{s.t.} \quad \forall i:\ y_i\left(\sum_{m=1}^{M} \boldsymbol{w}_m' \psi_m(\boldsymbol{x}_i) + b\right) \geq 1 - \xi_i\ ; \quad \boldsymbol{\xi} \geq \mathbf{0}\ ; \quad \boldsymbol{\theta} \geq \mathbf{0}\ ; \quad ||\boldsymbol{\theta}||_p^p \leq 1.$$

Applying Lagrange's theorem incorporates the constraints into the objective by introducing non-negative Lagrangian multipliers $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^n, \boldsymbol{\gamma} \in \mathbb{R}^M, \delta \in \mathbb{R}$ (including a pre-factor of $\frac{1}{p}$ for the δ-Term). Resubstitution of optimality conditions w.r.t. to $\boldsymbol{w}$, b, $\boldsymbol{\xi}$, and $\boldsymbol{\theta}$ removes the dependency of the Lagrangian on the primal variables. After some additional algebra (e.g., the terms associated with $\boldsymbol{\gamma}$ cancel), the Lagrangian can be written as

$$\mathcal{L} = \mathbf{1}^\top \boldsymbol{\alpha} - \frac{1}{p}\delta - \frac{p-1}{p}\delta^{-\frac{1}{p-1}} \left(\sum_{m=1}^{M} \left(\frac{1}{2}\boldsymbol{\alpha}^\top Q_m \boldsymbol{\alpha} \right)^{\frac{p}{p-1}} \right), \tag{7}$$

where $Q_m = \text{diag}(\boldsymbol{y}) K_m \text{diag}(\boldsymbol{y})$. Eq. (7) now has to be maximized w.r.t. to the dual variables $\boldsymbol{\alpha}, \delta$, subject to $\boldsymbol{\alpha}^\top \boldsymbol{y} = 0$, $0 \leq \alpha_i \leq C$ for $1 \leq i \leq n$, and $\delta \geq 0$. Let us ignore for a moment the non-negativity $\delta \geq 0$ and solve $\partial \mathcal{L} / \partial \delta = 0$ for the unbounded δ. Setting the partial derivative to zero yields

$$\delta = \left(\sum_{m=1}^{M} \left(\frac{1}{2}\boldsymbol{\alpha}^\top Q_m \boldsymbol{\alpha} \right)^{\frac{p}{p-1}} \right)^{\frac{p-1}{p}}. \tag{8}$$

Interestingly, at optimality, we always have $\delta \geq 0$ because the quadratic term in $\boldsymbol{\alpha}$ is non-negative. Plugging the optimal δ into Eq. (7), we arrive at the following optimization problem which solely depends on $\boldsymbol{\alpha}$.

$$\max_{\boldsymbol{\alpha}} \quad \mathbf{1}^\top \boldsymbol{\alpha} - \frac{1}{2} \left(\sum_{m=1}^{M} \left(\boldsymbol{\alpha}^\top Q_m \boldsymbol{\alpha} \right)^{\frac{p}{p-1}} \right)^{\frac{p-1}{p}} \qquad \text{s.t.} \quad \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}; \quad \boldsymbol{\alpha}^\top \boldsymbol{y} = 0. \tag{9}$$

In the limit $p \to \infty$, the above problem reduces to the SVM dual (with $Q = \sum_m Q_m$), while $p \to 1$ gives rise to a QCQP ℓ_1-MKL variant. However, optimizing the dual efficiently is difficult and will cause numerical problems in the limits $p \to 1$ and $p \to \infty$.

2.4 Two Efficient Second-Order Optimization Strategies

Many recent MKL solvers (e.g., [19, 24, 26]) are based on wrapping linear programs around SVMs. From an optimization standpoint, our work is most closely related to the SILP approach [21] and the simpleMKL method [19, 24]. Both of these methods also aim at efficient large-scale MKL algorithms. The two alternative approaches proposed for ℓ_p-norm MKL proposed in this paper are largely inspired by these methods and extend them in two aspects: customization to arbitrary norms and a tight coupling with minor iterations of an SVM solver, respectively.

Our first strategy interleaves maximizing the Lagrangian of (6) w.r.t. $\boldsymbol{\alpha}$ with minor precision and Newton descent on $\boldsymbol{\theta}$. For the second strategy, we devise a semi-infinite convex program, which we solve by column generation with nested sequential quadratically constrained linear programming (SQCLP). In both cases, the maximization step w.r.t. $\boldsymbol{\alpha}$ is performed by chunking optimization with minor iterations. The Newton approach can be applied without a common purpose QCQP solver, however, convergence can only be guaranteed for the SQCLP [8].

2.4.1 Newton Descent

For a Newton descent on the mixing coefficients, we first compute the partial derivatives

$$\frac{\partial \mathcal{L}}{\partial \theta_m} = \underbrace{-\frac{1}{2}\frac{\boldsymbol{w}_m^\top \boldsymbol{w}_m}{\theta_m^2} + \delta\theta_m^{p-1}}_{=:\nabla_{\theta_m}} \qquad \text{and} \qquad \frac{\partial^2 \mathcal{L}}{\partial^2 \theta_m} = \underbrace{\frac{\boldsymbol{w}_m^\top \boldsymbol{w}_m}{\theta_m^3} + (p-1)\delta\theta_m^{p-2}}_{=:h_m}$$

of the original Lagrangian. Fortunately, the Hessian H is diagonal, i.e. given by $H = \text{diag}(\boldsymbol{h})$. The m-th element s_m of the corresponding Netwon step, defined as $\boldsymbol{s} := -H^{-1}\nabla_{\boldsymbol{\theta}}$, is thus computed by

$$s_m = \frac{\frac{1}{2}\theta_m ||\boldsymbol{w}_m||^2 - \delta\theta_m^{p+2}}{||\boldsymbol{w}_m||^2 + (p-1)\delta\theta_m^{p+1}},$$

where δ is defined in Eq. (8). However, a Newton step $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \boldsymbol{s}$ might lead to non-positive $\boldsymbol{\theta}$. To avoid this awkward situation, we take the Newton steps in the space of $\log(\boldsymbol{\theta})$ by adjusting the derivatives according to the chain rule. We obtain

$$\log(\theta_m^{t+1}) \quad = \quad \log(\theta_m^t) - \frac{\nabla_{\theta_m}^t / \theta_m^t}{h_m^t/(\theta_m^t)^2 - \nabla_{\theta_m}^t/(\theta_m^t)^2} \quad , \tag{10}$$

which corresponds to multiplicative update of $\boldsymbol{\theta}$:

$$\theta_m^{t+1} \quad = \quad \theta_m^t \cdot \exp\left(\frac{\nabla_{\theta_m}^t \theta_m^t}{\nabla_{\theta_m}^t - h_m^t}\right) \quad . \tag{11}$$

Furthermore we additionally enhance the Newton step by a line search.

2.4.2 Cutting Planes

In order to obtain an alternative optimization strategy, we fix $\boldsymbol{\theta}$ and build the partial Lagrangian w.r.t. all other primal variables $\boldsymbol{w}$, b, $\boldsymbol{\xi}$. The derivation is analogous to [18, 27] and we omit details for lack of space. The resulting dual problem is a min-max problem of the form

$$\begin{aligned} \min_{\boldsymbol{\theta}} \ \max_{\boldsymbol{\alpha}} \quad & \mathbf{1}^\top \boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^\top \sum_{m=1}^{M} \theta_m Q_m \boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{0} \le \boldsymbol{\alpha} \le C\mathbf{1}; \quad \boldsymbol{y}^\top \boldsymbol{\alpha} = 0; \quad \boldsymbol{\theta} \ge 0; \quad \|\boldsymbol{\theta}\|_p^p \le 1. \end{aligned}$$

The above optimization problem is a *saddle point problem* and can be solved by alternating $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ optimization step. While the former can simply be carried out by a support vector machine for a fixed mixture $\boldsymbol{\theta}$, the latter has been optimized for $p = 1$ by reduced gradients [18].

We take a different approach and translate the min-max problem into an equivalent semi-infinite program (SIP) as follows. Denote the value of the target function by $t(\boldsymbol{\alpha}, \boldsymbol{\theta})$ and suppose $\boldsymbol{\alpha}^*$ is optimal. Then, according to the max-min inequality [5], we have $t(\boldsymbol{\alpha}^*, \boldsymbol{\theta}) \ge t(\boldsymbol{\alpha}, \boldsymbol{\theta})$ for all $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$. Hence, we can equivalently minimize an upper bound η on the optimal value and arrive at

$$\min_{\eta, \boldsymbol{\theta}} \quad \eta \quad \text{s.t.} \quad \eta \ge \mathbf{1}^\top \boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^\top \sum_{m=1}^{M} \theta_m Q_m \boldsymbol{\alpha} \tag{12}$$

for all $\boldsymbol{\alpha} \in \mathbb{R}^n$ with $\mathbf{0} \le \boldsymbol{\alpha} \le C\mathbf{1}$, and $\boldsymbol{y}^\top \boldsymbol{\alpha} = 0$ as well as $\|\boldsymbol{\theta}\|_p^p \le 1$ and $\boldsymbol{\theta} \ge \mathbf{0}$.

[21] optimize the above SIP for $p \ge 1$ with interleaving cutting plane algorithms. The solution of a quadratic program (here the regular SVM) generates the most strongly violated constraint for the actual mixture $\boldsymbol{\theta}$. The optimal $(\boldsymbol{\theta}^*, \eta)$ is then identified by solving a linear program with respect to the set of active constraints. The optimal mixture is then used for computing a new constraint and so on.

Unfortunately, for $p > 1$, a non-linearity is introduced by requiring $\|\boldsymbol{\theta}\|_p^p \le 1$ and such constraint is unlikely to be found in standard optimization toolboxes that often handle only linear and quadratic constraints. As a remedy, we propose to approximate $\|\boldsymbol{\theta}\|_p^p \le 1$ by sequential second-order Taylor expansion of the form

$$\|\boldsymbol{\theta}\|_p^p \;\approx\; 1 + \frac{p(p-3)}{2} - \sum_{m=1}^{M} p(p-2)(\tilde{\theta}_m)^{p-1}\, \theta_m \;+ \frac{p(p-1)}{2} \sum_{m=1}^{M} \tilde{\theta}_m^{p-2}\, \theta_m^2,$$

where $\boldsymbol{\theta}^p$ is defined element-wise, that is $\boldsymbol{\theta}^p := (\theta_1^p, ..., \theta_M^p)$. The sequence $(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \cdots)$ is initialized with a uniform mixture satisfying $\|\boldsymbol{\theta}_0\|_p^p = 1$ as a starting point. Successively $\boldsymbol{\theta}_{t+1}$ is computed using $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}_t$. Note that the quadratic term in the approximation is diagonal wherefore the subsequent quadratically constrained problem can be solved efficiently. Finally note, that this approach can be further sped-up by an additional projection onto the level-sets in the $\boldsymbol{\theta}$-optimization phase similar to [26]. In our case, the level-set projection is a convex quadratic problem with ℓ_p-norm constraints and can again be approximated by successive second-order Taylor expansions.

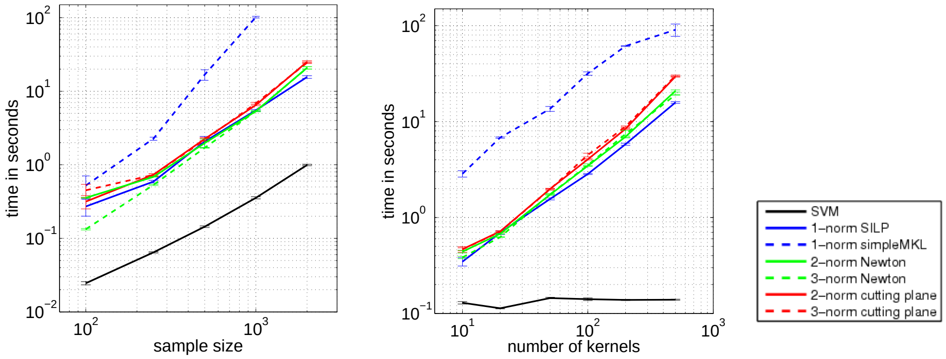

Figure 1: Execution times of SVM Training, ℓ_p-norm MKL based on interleaved optimization via the Newton, the cutting plane algorithm (CPA), and the SimpleMKL wrapper. (left) Training using fixed number of 50 kernels varying training set size. (right) For 500 examples and varying numbers of kernels. Our proposed Newton and CPA obtain speedups of over an order of magnitude. Notice the tiny error bars.

3 Computational Experiments

In this section we study non-sparse MKL in terms of efficiency and accuracy.[1] We apply the method of [21] for ℓ_1-norm results as it is contained as a special case of our cutting plane strategy. We write ℓ_∞-norm MKL for a regular SVM with the unweighted-sum kernel $K = \sum_m K_m$.

3.1 Execution Time

We demonstrate the efficiency of our implementations of non-sparse MKL. We experiment on the MNIST data set where the task is to separate odd vs. even digits. We compare our ℓ_p-norm MKL with two methods for ℓ_1-norm MKL, simpleMKL [19] and SILP-based chunking [21], and to SVMs using the unweighted-sum kernel (ℓ_∞-norm MKL) as additional baseline. We optimize all methods up to a precision of 10^{-3} for the outer SVM-ε and 10^{-5} for the "inner" SIP precision and computed relative duality gaps. To provide a fair stopping criterion to simpleMKL, we set the stopping criterion of simpleMKL to the relative duality gap of its ℓ_1-norm counterpart. This way, the deviations of relative objective values of ℓ_1-norm MKL variants are guaranteed to be smaller than 10^{-4}. SVM trade-off parameters are set to $C = 1$ for all methods.

Figure 1 (left) displays the results for varying sample sizes and 50 precomputed Gaussian kernels with different bandwidths. Error bars indicate standard error over 5 repetitions. Unsurprisingly, the SVM with the unweighted-sum kernel is the fastest method. Non-sparse MKL scales similarly as ℓ_1-norm chunking; the Newton strategy (Section 2.4.1) is slightly faster than the cutting plane variant (Section 2.4.2) that needs additional Taylor expansions within each $\boldsymbol{\theta}$-step. SimpleMKL suffers from training an SVM to full precision for each gradient evaluation and performs worst.[2]

Figure 1 (right) shows the results for varying the number of precomputed RBF kernels for a fixed sample size of 500. The SVM with the unweighted-sum kernel is hardly affected by this setup and performs constantly. The ℓ_1-norm MKL by [21] handles the increasing number of kernels best and is the fastest MKL method. Non-sparse approaches to MKL show reasonable run-times, the Newton-based ℓ_p-norm MKL being again slightly faster than its peer. Simple MKL performs again worst. Overall, our proposed Newton and cutting plane based optimization strategies achieve a speedup of often more than one order of magnitude.

3.2 Protein Subcellular Localization

The prediction of the subcellular localization of proteins is one of the rare empirical success stories of ℓ_1-norm-regularized MKL [17, 27]: after defining 69 kernels that capture diverse aspects of

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

[2] SimpleMKL could not be evaluated for 2000 instances (ran out of memory on a 4GB machine).

Table 1: Results for Protein Subcellular Localization

ℓ_p-norm	1	$32/31$	$16/15$	$8/7$	$4/3$	2	4	∞
1 - MCC [%]	9.13	9.12	9.64	9.84	9.56	10.18	10.08	10.41

protein sequences, ℓ_1-norm-MKL could raise the predictive accuracy significantly above that of the unweighted sum of kernels (thereby also improving on established prediction systems for this problem). Here we investigate the performance of non-sparse MKL.

We download the kernel matrices of the dataset `plant`[3] and follow the experimental setup of [17] with the following changes: instead of a genuine multiclass SVM, we use the 1-vs-rest decomposition; instead of performing cross-validation for model selection, we report results for the best models, as we are only interested in the relative performance of the MKL regularizers. Specifically, for each $C \in \{1/32, 1/8, 1/2, 1, 2, 4, 8, 32, 128\}$, we compute the average Mathews correlation coefficient (MCC) on the test data. For each norm, the best average MCC is recorded. Table 1 shows the averages over several splits of the data.

The results indicate that, indeed, with proper choice of a non-sparse regularizer, the accuracy of ℓ_1-norm can be recovered. This is remarkable, as this dataset is particular in that it fullfills the rare condition that ℓ_1-norm MKL performs better than ℓ_∞-norm MKL. In other words, selecting these data may imply a bias towards ℓ_1-norm. Nevertheless our novel non-sparse MKL can keep up with this, essentially by approximating ℓ_1-norm.

3.3 Gene Start Recognition

This experiment aims at detecting transcription start sites (TSS) of RNA Polymerase II binding genes in genomic DNA sequences. Accurate detection of the transcription start site is crucial to identify genes and their promoter regions and can be regarded as a first step in deciphering the key regulatory elements in the promoter region that determine transcription. For our experiments we use the dataset from [22] which contains a curated set of 8,508 TSS annotated genes built from dbTSS version 4 [23] and refseq genes. These are translated into positive training instances by extracting windows of size $[-1000, +1000]$ around the TSS. Similar to [4], 85,042 negative instances are generated from the interior of the gene using the same window size.

Following [22], we employ five different kernels representing the TSS signal (weighted degree with shift), the promoter (spectrum), the 1st exon (spectrum), angles (linear), and energies (linear). Optimal kernel parameters are determined by model selection in [22]. Every kernel is normalized such that all points have unit length in feature space. We reserve 13,000 and 20,000 randomly drawn instances for holdout and test sets, respectively, and use the remaining 60,000 as the training pool. Figure 2 shows test errors for varying training set sizes drawn from the pool; training sets of the same size are disjoint. Error bars indicate standard errors of repetitions for small training set sizes.

Regardless of the sample size, ℓ_1-MKL is significantly outperformed by the sum-kernel. On the contrary, non-sparse MKL significantly achieves higher AUC values than the ℓ_∞-MKL for sample sizes up to 20k. The scenario is well suited for ℓ_2-norm MKL which performs best. Finally, for 60k training instances, all methods but ℓ_1-norm MKL yield the same performance. Again, the superior performance of non-sparse MKL is remarkable, and of significance for the application domain: the method using the unweighted sum of kernels [22] has recently been confirmed to be the leading in a comparison of 19 state-of-the-art promoter prediction programs [1], and our experiments suggest that its accuracy can be further elevated by non-sparse MKL.

4 Conclusion and Discussion

We presented an efficient and accurate approach to non-sparse multiple kernel learning and showed that our ℓ_p-norm MKL can be motivated as Tikhonov and Ivanov regularization of the mixing coefficients, respectively. Applied to previous MKL research, our result allows for a unified view as so far seemingly different approaches turned out to be equivalent. Furthermore, we devised two efficient approaches to non-sparse multiple kernel learning for arbitrary ℓ_p-norms, $p > 1$. The resulting

[3] from `http://www.fml.tuebingen.mpg.de/raetsch/suppl/protsubloc/`

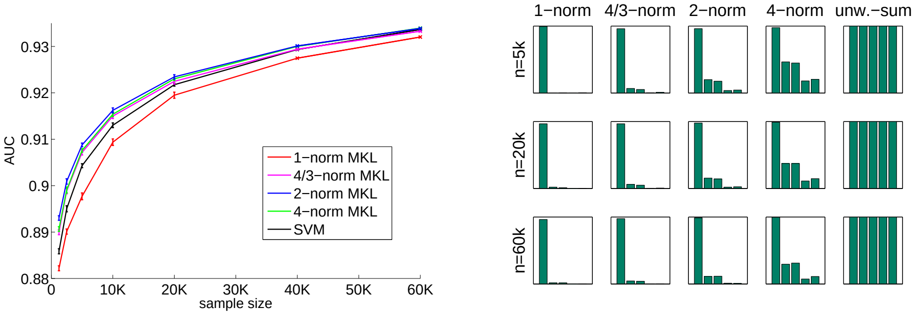

Figure 2: Left: Area under ROC curve (AUC) on test data for TSS recognition as a function of the training set size. Notice the tiny bars indicating standard errors w.r.t. repetitions on disjoint training sets. Right: Corresponding kernel mixtures. For $p = 1$ consistent sparse solutios are obtained while the optimal $p = 2$ distributes wheights on the weighted degree and the 2 spectrum kernels in good agreement to [22].

optimization strategies are based on semi-infinite programming and Newton descent, both interleaved with chunking-based SVM training. Execution times moreover revealed that our interleaved optimization vastly outperforms commonly used wrapper approaches.

We would like to note that there is a certain preference/obsession for sparse models in the scientific community due to various reasons. The present paper, however, shows clearly that sparsity by itself is not the ultimate virtue to be strived for. Rather on the contrary: non-sparse model may improve quite impressively over sparse ones. The reason for this is less obvious and its theoretical exploration goes well beyond the scope of its submissions. We remark nevertheless that some interesting asymptotic results exist that show *model selection consistency* of sparse MKL (or the closely related group lasso) [2, 14], in other words in the limit $n \to \infty$ MKL is guaranteed to find the correct subset of kernels. However, also the rate of convergence to the true estimator needs to be considered, thus we conjecture that the rate slower than $\sqrt{n}$ which is common to sparse estimators [11] may be one of the reasons for finding excellent (nonasymptotic) results in non-sparse MKL. In addition to the convergence rate the variance properties of MKL estimators may play an important role to elucidate the performance seen in our various simulation experiments.

Intuitively speaking, we observe clearly that in some cases all features even though they may contain redundant information are to be kept, since putting their contributions to zero does not improve prediction. I.e. all of them are informative to our MKL models. Note however that this result is also class specific, i.e. for some classes we may sparsify. Cross-validation based model building that includes the choice of p will however inevitably tell us which classes should be treated sparse and which non-sparse.

Large-scale experiments on TSS recognition even raised the bar for ℓ_1-norm MKL: non-sparse MKL proved consistently better than its sparse counterparts which were outperformed by an unweighted-sum kernel. This exemplifies how the unprecedented combination of accuracy and scalability of our MKL approach and methods paves the way for progress in other real world applications of machine learning.

Authors' Contributions

The authors contributed in the following way: MK and UB had the initial idea. MK, UB, SS, and AZ each contributed substantially to both mathematical modelling, design and implementation of algorithms, conception and execution of experiments, and writing of the manuscript. PL had some shares in the initial phase and KRM contributed to the text. Most of the work was done at previous affiliations of several authors: Fraunhofer Institute FIRST (Berlin), Technical University Berlin, and the Friedrich Miescher Laboratory (Tübingen).

Acknowledgments

This work was supported in part by the German BMBF grant REMIND (FKZ 01-IS07007A) and by the European Community under the PASCAL2 Network of Excellence (ICT-216886).

References

[1] T. Abeel, Y. V. de Peer, and Y. Saeys. Towards a gold standard for promoter prediction evaluation. *Bioinformatics*, 2009.

[2] F. R. Bach. Consistency of the group lasso and multiple kernel learning. *J. Mach. Learn. Res.*, 9:1179–1225, 2008.

[3] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proc. 21st ICML*. ACM, 2004.

[4] V. B. Bajic, S. L. Tan, Y. Suzuki, and S. Sugano. Promoter prediction analysis on the whole human genome. *Nature Biotechnology*, 22(11):1467–1473, 2004.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambrigde University Press, Cambridge, UK, 2004.

[6] O. Chapelle and A. Rakotomamonjy. Second order optimization of kernel parameters. In *Proc. of the NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels*, 2008.

[7] C. Cortes, A. Gretton, G. Lanckriet, M. Mohri, and A. Rostamizadeh. Proceedings of the NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels, 2008.

[8] R. Hettich and K. O. Kortanek. Semi-infinite programming: theory, methods, and applications. *SIAM Rev.*, 35(3):380–429, 1993.

[9] S. Ji, L. Sun, R. Jin, and J. Ye. Multi-label multiple kernel learning. In *Advances in Neural Information Processing Systems*, 2009.

[10] G. Lanckriet, N. Cristianini, L. E. Ghaoui, P. Bartlett, and M. I. Jordan. Learning the kernel matrix with semi-definite programming. *JMLR*, 5:27–72, 2004.

[11] H. Leeb and B. M. Pötscher. Sparse estimators and the oracle property, or the return of hodges' estimator. *Journal of Econometrics*, 142:201–211, 2008.

[12] C. Longworth and M. J. F. Gales. Combining derivative and parametric kernels for speaker verification. *IEEE Transactions in Audio, Speech and Language Processing*, 17(4):748–757, 2009.

[13] C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6:1099–1125, 2005.

[14] Y. Nardi and A. Rinaldo. On the asymptotic properties of the group lasso estimator for linear models. *Electron. J. Statist.*, 2:605–633, 2008.

[15] S. Olhede, M. Pontil, and J. Shawe-Taylor. Proceedings of the PASCAL2 Workshop on Sparsity in Machine Learning and Statistics, 2009.

[16] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[17] C. S. Ong and A. Zien. An Automated Combination of Kernels for Predicting Protein Subcellular Localization. In *Proc. of the 8th Workshop on Algorithms in Bioinformatics*, 2008.

[18] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. More efficiency in multiple kernel learning. In *ICML*, pages 775–782, 2007.

[19] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[20] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[21] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large Scale Multiple Kernel Learning. *Journal of Machine Learning Research*, 7:1531–1565, July 2006.

[22] S. Sonnenburg, A. Zien, and G. Rätsch. ARTS: Accurate Recognition of Transcription Starts in Human. *Bioinformatics*, 22(14):e472–e480, 2006.

[23] Y. Suzuki, R. Yamashita, K. Nakai, and S. Sugano. dbTSS: Database of human transcriptional start sites and full-length cDNAs. *Nucleic Acids Research*, 30(1):328–331, 2002.

[24] M. Szafranski, Y. Grandvalet, and A. Rakotomamonjy. Composite kernel learning. In *Proceedings of the International Conference on Machine Learning*, 2008.

[25] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *IEEE 11th International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

[26] Z. Xu, R. Jin, I. King, and M. Lyu. An extended level method for efficient multiple kernel learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1825–1832. 2009.

[27] A. Zien and C. S. Ong. Multiclass multiple kernel learning. In *Proceedings of the 24th international conference on Machine learning (ICML)*, pages 1191–1198. ACM, 2007.