Journal of Machine Learning Research 12 (2011) 953-997 Submitted 2/10; Revised 10/10; Published 3/11

ℓ_p-Norm Multiple Kernel Learning

Marius Kloft[*] MKLOFT@CS.BERKELEY.EDU
University of California
Computer Science Division
Berkeley, CA 94720-1758, USA

Ulf Brefeld BREFELD@YAHOO-INC.COM
Yahoo! Research
Avinguda Diagonal 177
08018 Barcelona, Spain

Sören Sonnenburg[†] SOEREN.SONNENBURG@TU-BERLIN.DE
Technische Universität Berlin
Franklinstr. 28/29
10587 Berlin, Germany

Alexander Zien[‡] ZIEN@LIFEBIOSYSTEMS.COM
LIFE Biosystems GmbH
Belfortstraße 2
69115 Heidelberg, Germany

Editor: Francis Bach

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 and scalability. Unfortunately, this ℓ_1-norm MKL is rarely observed to outperform trivial baselines in practical applications. To allow for robust kernel mixtures that generalize well, we extend MKL to arbitrary norms. We devise new insights on the connection between several existing MKL formulations and develop two efficient *interleaved* optimization strategies for arbitrary norms, that is ℓ_p-norms with $p \geq 1$. This interleaved optimization is much faster than the commonly used wrapper approaches, as demonstrated on several data sets. A theoretical analysis and an experiment on controlled artificial data shed light on the appropriateness of sparse, non-sparse and ℓ_∞-norm MKL in various scenarios. Importantly, empirical applications of ℓ_p-norm MKL to three real-world problems from computational biology show that non-sparse MKL achieves accuracies that surpass the state-of-the-art.

Data sets, source code to reproduce the experiments, implementations of the algorithms, and further information are available at `http://doc.ml.tu-berlin.de/nonsparse_mkl/`.

Keywords: multiple kernel learning, learning kernels, non-sparse, support vector machine, convex conjugate, block coordinate descent, large scale optimization, bioinformatics, generalization bounds, Rademacher complexity

---

∗. Also at Machine Learning Group, Technische Universität Berlin, 10587 Berlin, Germany.

†. Parts of this work were done while SS was at the Friedrich Miescher Laboratory, Max Planck Society, 72076 Tübingen, Germany.

‡. Most contributions by AZ were done at the Fraunhofer Institute FIRST, 12489 Berlin, Germany.

1. Introduction

Kernels allow to decouple machine learning from data representations. Finding an appropriate data representation via a kernel function immediately opens the door to a vast world of powerful machine learning models (e.g., Schölkopf and Smola, 2002) with many efficient and reliable off-the-shelf implementations. This has propelled the dissemination of machine learning techniques to a wide range of diverse application domains.

Finding an appropriate data abstraction—or even engineering *the best* kernel—for the problem at hand is not always trivial, though. Starting with cross-validation (Stone, 1974), which is probably the most prominent approach to general model selection, a great many approaches to selecting the right kernel(s) have been deployed in the literature.

Kernel target alignment (Cristianini et al., 2002; Cortes et al., 2010b) aims at learning the entries of a kernel matrix by using the outer product of the label vector as the ground-truth. Chapelle et al. (2002) and Bousquet and Herrmann (2002) minimize estimates of the generalization error of support vector machines (SVMs) using a gradient descent algorithm over the set of parameters. Ong et al. (2005) study hyperkernels on the space of kernels and alternative approaches include selecting kernels by DC programming (Argyriou et al., 2008) and semi-infinite programming (Özögür-Akyüz and Weber, 2008; Gehler and Nowozin, 2008). Although finding non-linear kernel mixtures (Gönen and Alpaydin, 2008; Varma and Babu, 2009) generally results in non-convex optimization problems, Cortes et al. (2009b) show that convex relaxations may be obtained for special cases.

However, learning arbitrary kernel combinations is a problem too general to allow for a general optimal solution—by focusing on a restricted scenario, it is possible to achieve guaranteed optimality. In their seminal work, Lanckriet et al. (2004) consider training an SVM along with optimizing the linear combination of several positive semi-definite matrices, $K = \sum_{m=1}^{M} \theta_m K_m$, subject to the trace constraint $\mathrm{tr}(K) \leq c$ and requiring a valid combined kernel $K \succeq 0$. This spawned the new field of *multiple kernel learning* (MKL), the automatic combination of several kernel functions. Lanckriet et al. (2004) show that their specific version of the MKL task can be reduced to a convex optimization problem, namely a semi-definite programming (SDP) optimization problem. Though convex, however, the SDP approach is computationally too expensive for practical applications. Thus much of the subsequent research focuses on devising more efficient optimization procedures.

One conceptual milestone for developing MKL into a tool of practical utility is simply to constrain the mixing coefficients θ to be non-negative: by obviating the complex constraint $K \succeq 0$, this small restriction allows to transform the optimization problem into a quadratically constrained program, hence drastically reducing the computational burden. While the original MKL objective is stated and optimized in dual space, alternative formulations have been studied. For instance, Bach et al. (2004) found a corresponding primal problem, and Rubinstein (2005) decomposed the MKL problem into a min-max problem that can be optimized by mirror-prox algorithms (Nemirovski, 2004). The min-max formulation has been independently proposed by Sonnenburg et al. (2005). They use it to recast MKL training as a semi-infinite linear program. Solving the latter with column generation (e.g., Nash and Sofer, 1996) amounts to repeatedly training an SVM on a mixture kernel while iteratively refining the mixture coefficients θ. This immediately lends itself to a convenient implementation by a wrapper approach. These wrapper algorithms directly benefit from efficient SVM optimization routines (cf., Fan et al., 2005; Joachims, 1999) and are now commonly deployed in recent MKL solvers (e.g., Rakotomamonjy et al., 2008; Xu et al., 2009), thereby allowing for large-scale training (Sonnenburg et al., 2005, 2006a). However, the complete training of several

SVMs can still be prohibitive for large data sets. For this reason, Sonnenburg et al. (2005) also propose to interleave the SILP with the SVM training which reduces the training time drastically. Alternative optimization schemes include level-set methods (Xu et al., 2009) and second order approaches (Chapelle and Rakotomamonjy, 2008). Szafranski et al. (2010), Nath et al. (2009), and Bach (2009) study composite and hierarchical kernel learning approaches. Finally, Zien and Ong (2007) and Ji et al. (2009) provide extensions for multi-class and multi-label settings, respectively.

Today, there exist two major families of multiple kernel learning models. The first is characterized by Ivanov regularization (Ivanov et al., 2002) over the mixing coefficients (Rakotomamonjy et al., 2007; Zien and Ong, 2007). For the Tikhonov-regularized optimization problem (Tikhonov and Arsenin, 1977), there is an additional parameter controlling the regularization of the mixing coefficients (Varma and Ray, 2007).

All the above mentioned multiple kernel learning formulations promote *sparse* solutions in terms of the mixing coefficients. The desire for sparse mixtures originates in practical as well as theoretical reasons. First, sparse combinations are easier to interpret. Second, irrelevant (and possibly expensive) kernels functions do not need to be evaluated at testing time. Finally, sparseness appears to be handy also from a technical point of view, as the additional simplex constraint $\|\theta\|_1 \leq 1$ simplifies derivations and turns the problem into a linearly constrained program. Nevertheless, sparseness is not always beneficial in practice and sparse MKL is frequently observed to be outperformed by a regular SVM using an unweighted-sum kernel $K = \sum_m K_m$ (Cortes et al., 2008).

Consequently, despite all the substantial progress in the field of MKL, there still remains an unsatisfied need for an approach that is really useful for practical applications: a model that has a good chance of improving the accuracy (over a plain sum kernel) together with an implementation that matches today's standards (i.e., that can be trained on 10,000s of data points in a reasonable time). In addition, since the field has grown several competing MKL formulations, it seems timely to consolidate the set of models. In this article we argue that all of this is now achievable.

1.1 Outline of the Presented Achievements

On the theoretical side, we cast multiple kernel learning as a general regularized risk minimization problem for arbitrary convex loss functions, Hilbertian regularizers, and arbitrary norm-penalties on θ. We first show that the above mentioned Tikhonov and Ivanov regularized MKL variants are equivalent in the sense that they yield the same set of hypotheses. Then we derive a dual representation and show that a variety of methods are special cases of our objective. Our optimization problem subsumes state-of-the-art approaches to multiple kernel learning, covering sparse and non-sparse MKL by arbitrary p-norm regularization ($1 \leq p \leq \infty$) on the mixing coefficients as well as the incorporation of prior knowledge by allowing for non-isotropic regularizers. As we demonstrate, the p-norm regularization includes both important special cases (sparse 1-norm and plain sum ∞-norm) and offers the potential to elevate predictive accuracy over both of them.

With regard to the implementation, we introduce an appealing and efficient optimization strategy which grounds on an exact update in closed-form in the θ-step; hence rendering expensive semi-infinite and first- or second-order gradient methods unnecessary. By using proven working set optimization for SVMs, p-norm MKL can now be trained highly efficiently for all p; in particular, we outpace other current 1-norm MKL implementations. Moreover our implementation employs kernel caching techniques, which enables training on ten thousands of data points or thousands of kernels respectively. In contrast, most competing MKL software require all kernel matrices

to be stored completely in memory, which restricts these methods to small data sets with limited numbers of kernels. Our implementation is freely available within the SHOGUN machine learning toolbox available at `http://www.shogun-toolbox.org/`. See also our supplementary homepage: `http://doc.ml.tu-berlin.de/nonsparse_mkl/`.

Our claims are backed up by experiments on artificial and real world data sets representing diverse, relevant and challenging problems from the application domain of bioinformatics. Using artificial data, we investigate the impact of the p-norm on the test error as a function of the size of the true sparsity pattern. The real world problems include subcellular localization of proteins, transcription start site detection, and enzyme function prediction. The results demonstrate (i) that combining kernels is now tractable on large data sets, (ii) that it can provide cutting edge classification accuracy, and (iii) that depending on the task at hand, different kernel mixture regularizations are required for achieving optimal performance.

We also present a theoretical analysis of non-sparse MKL. We introduce a novel ℓ_1-to-ℓ_p conversion technique and use it to derive generalization bounds. Based on these, we perform a case study to compare an exemplary sparse with a non-sparse learning scenario. We show that in the sparse scenario $\ell_{p>1}$-norm MKL yields a strictly better generalization bound than ℓ_1-norm MKL, while in the non-sparse scenario it is the other way around.

The remainder is structured as follows. We derive non-sparse MKL in Section 2 and discuss relations to existing approaches in Section 3. Section 4.3 introduces the novel optimization strategy and its implementation. We report on theoretical results in Section 5 and on our empirical results in Section 6. Section 7 concludes.

1.1.1 RELATED WORK

A basic version of this work appeared in NIPS 2009 (Kloft et al., 2009a). The present article additionally offers a more general and complete derivation of the main optimization problem, exemplary applications thereof, a simple algorithm based on a closed-form solution, technical details of the implementation, a theoretical analysis, and additional experimental results. Parts of Section 5 are based on Kloft et al. (2010) the present analysis however extends the previous publication by a novel conversion technique, an illustrative case study, tighter bounds, and an improved presentation.

In related papers, non-sparse MKL has been applied, extended, and further analyzed by several researchers since its initial publication in Kloft et al. (2008), Cortes et al. (2009a), and Kloft et al. (2009a): Varma and Babu (2009) derive a projected gradient-based optimization method for ℓ_2-norm MKL. Yu et al. (2010) present a more general dual view of ℓ_2-norm MKL and show advantages of ℓ_2-norm over an unweighted-sum kernel SVM on six bioinformatics data sets. Cortes et al. (2010a) provide generalization bounds for ℓ_1- and $\ell_{p\leq 2}$-norm MKL. The analytical optimization method presented in this paper was independently and in parallel discovered by Xu et al. (2010) and has also been studied in Roth and Fischer (2007) and Ying et al. (2009) for ℓ_1-norm MKL, and in Szafranski et al. (2010) and Nath et al. (2009) for composite kernel learning on small and medium scales.

2. Multiple Kernel Learning—A Unifying View

In this section we cast multiple kernel learning into a unified framework: we present a regularized loss minimization formulation with additional norm constraints on the kernel mixing coefficients.

We show that it comprises many popular MKL variants currently discussed in the literature, including seemingly different ones.

We derive generalized dual optimization problems without making specific assumptions on the norm regularizers or the loss function, beside that the latter is convex. As a special case we derive ℓ_p-norm MKL in Section 4. In addition, our formulation covers binary classification and regression tasks and can easily be extended to multi-class classification and structural learning settings using appropriate convex loss functions and joint kernel extensions (cf. Section 3). Prior knowledge on kernel mixtures and kernel asymmetries can be incorporated by non-isotropic norm regularizers.

2.1 Preliminaries

We begin with reviewing the classical supervised learning setup. Given a labeled sample $\mathcal{D} = \{(x_i, y_i)\}_{i=1\ldots,n}$, where the x_i lie in some input space $\mathcal{X}$ and $y_i \in \mathcal{Y} \subset \mathbb{R}$, the goal is to find a hypothesis $h \in H$, that generalizes well on new and unseen data. Regularized risk minimization returns a minimizer h^*,

$$h^* \in \operatorname{argmin}_h \mathrm{R}_{\mathrm{emp}}(h) + \lambda\Omega(h),$$

where $\mathrm{R}_{\mathrm{emp}}(h) = \frac{1}{n}\sum_{i=1}^{n} V(h(x_i), y_i)$ is the empirical risk of hypothesis h w.r.t. a convex loss function $V : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$, $\Omega : H \to \mathbb{R}$ is a regularizer, and $\lambda > 0$ is a trade-off parameter. We consider linear models of the form

$$h_{\tilde{w},b}(x) = \langle \tilde{w}, \psi(x) \rangle + b, \tag{1}$$

together with a (possibly non-linear) mapping $\psi : \mathcal{X} \to \mathcal{H}$ to a Hilbert space $\mathcal{H}$ (e.g., Schölkopf et al., 1998; Müller et al., 2001) and constrain the regularization to be of the form $\Omega(h) = \frac{1}{2}\|\tilde{w}\|_2^2$ which allows to kernelize the resulting models and algorithms. We will later make use of kernel functions $k(x, x') = \langle \psi(x), \psi(x') \rangle_{\mathcal{H}}$ to compute inner products in $\mathcal{H}$.

2.2 Regularized Risk Minimization 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 $\mathcal{H}_m$. Convex approaches to multiple kernel learning consider linear kernel mixtures $k_\theta = \sum \theta_m k_m$, $\theta_m \geq 0$. Compared to Equation (1), the primal model for learning with multiple kernels is extended to

$$h_{\tilde{w},b,\theta}(x) = \sum_{m=1}^{M} \sqrt{\theta_m} \langle \tilde{w}_m, \psi_m(x) \rangle_{\mathcal{H}_m} + b = \langle \tilde{w}, \psi_\theta(x) \rangle_{\mathcal{H}} + b$$

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

In learning with multiple kernels we aim at minimizing the loss on the training data w.r.t. the optimal kernel mixture $\sum_{m=1}^{M} \theta_m k_m$ in addition to regularizing θ to avoid overfitting. Hence, in terms of regularized risk minimization, the optimization problem becomes

$$\inf_{\tilde{w},b,\theta:\theta\geq\mathbf{0}} \quad \frac{1}{n}\sum_{i=1}^{n} V\left(\sum_{m=1}^{M} \sqrt{\theta_m}\langle \tilde{w}_m, \psi_m(x_i)\rangle_{\mathcal{H}_m} + b,\ y_i\right) + \frac{\lambda}{2}\sum_{m=1}^{M} \|\tilde{w}_m\|_{\mathcal{H}_m}^2 + \tilde{\mu}\tilde{\Omega}[\theta], \tag{2}$$

for $\tilde{\mu} > 0$. Note that the objective value of Equation (2) is an upper bound on the training error. Previous approaches to multiple kernel learning employ regularizers of the form $\tilde{\Omega}(\theta) = \|\theta\|_1$ to promote sparse kernel mixtures. In contrast, we propose to use convex regularizers of the form $\tilde{\Omega}(\theta) = \|\theta\|^2$, where $\|\cdot\|^2$ is an arbitrary norm in $\mathbb{R}^M$, possibly allowing for non-sparse solutions and the incorporation of prior knowledge. The non-convexity arising from the $\sqrt{\theta_m}\tilde{w}_m$ product in the loss term of Equation (2) is not inherent and can be resolved by substituting $w_m \leftarrow \sqrt{\theta_m}\tilde{w}_m$. Furthermore, the regularization parameter and the sample size can be decoupled by introducing $\tilde{C} = \frac{1}{n\tilde{\lambda}}$ (and adjusting $\mu \leftarrow \frac{\tilde{\mu}}{\tilde{\lambda}}$) which has favorable scaling properties in practice. We obtain the following convex optimization problem (Boyd and Vandenberghe, 2004) that has also been considered by Varma and Ray (2007) for hinge loss and an ℓ_1-norm regularizer

$$\inf_{w,b,\theta:\theta\geq\mathbf{0}} \quad \tilde{C}\sum_{i=1}^{n} V\left(\sum_{m=1}^{M}\langle w_m, \psi_m(x_i)\rangle_{\mathcal{H}_m} + b,\, y_i\right) + \frac{1}{2}\sum_{m=1}^{M}\frac{\|w_m\|^2_{\mathcal{H}_m}}{\theta_m} + \mu\|\theta\|^2, \tag{3}$$

where we use the convention that $\frac{t}{0} = 0$ if $t = 0$ and ∞ otherwise.

An alternative approach has been studied by Rakotomamonjy et al. (2007) and Zien and Ong (2007), again using hinge loss and ℓ_1-norm. They upper bound the value of the regularizer $\|\theta\|_1 \leq 1$ and incorporate the regularizer as an additional constraint into the optimization problem. For $C > 0$ and hinge loss, they arrive at the following problem which is the primary object of investigation in this paper.

2.2.1 GENERAL PRIMAL MKL OPTIMIZATION PROBLEM

$$\begin{aligned} \inf_{w,b,\theta:\theta\geq\mathbf{0}} \quad & C\sum_{i=1}^{n} V\Big(\sum_{m=1}^{M}\langle w_m, \psi_m(x_i)\rangle_{\mathcal{H}_m} + b,\, y_i\Big) + \frac{1}{2}\sum_{m=1}^{M}\frac{\|w_m\|^2_{\mathcal{H}_m}}{\theta_m} \\ \text{s.t.} \quad & \|\theta\|^2 \leq 1. \end{aligned} \tag{4}$$

It is important to note here that, while the Tikhonov regularization in (3) has *two* regularization parameters (C and μ), the above Ivanov regularization (4) has only *one* (C only). Our first contribution shows that, despite the additional regularization parameter, both MKL variants are equivalent, in the sense that traversing the regularization paths yields the same binary classification functions.

Theorem 1 *Let $\|\cdot\|$ be a norm on $\mathbb{R}^M$ and V a convex loss function. Suppose for the optimal w^* in Optimization Problem* (4) *it holds $w^* \neq \mathbf{0}$. Then, for each pair $(\tilde{C}, \mu)$ there exists $C > 0$ such that for each optimal solution (w, b, θ) of Equation* (3) *using $(\tilde{C}, \mu)$, we have that $(w, b, \kappa\theta)$ is also an optimal solution of Optimization Problem* (4) *using C, and vice versa, where $\kappa > 0$ is a multiplicative constant.*

For the proof we need Prop. 12, which justifies switching from Ivanov to Tikhonov regularization, and back, if the regularizer is tight. We refer to Appendix A for the proposition and its proof.

Proof of Theorem 1 Let be $(\tilde{C}, \mu) > 0$. In order to apply Prop. 12 to (3), we show that condition (31) in Prop. 12 is satisfied, that is, that the regularizer is tight.

Suppose on the contrary, that Optimization Problem (4) yields the same infimum regardless of whether we require

$$\|\theta\|^2 \leq 1,$$

or not. Then this implies that in the optimal point we have $\sum_{m=1}^{M} \frac{\|w_m^*\|_2^2}{\theta_m^*} = 0$, hence,

$$\frac{\|w_m^*\|_2^2}{\theta_m^*} = 0, \ \ \forall\, m = 1, \ldots, M. \tag{5}$$

Since all norms on $\mathbb{R}^M$ are equivalent (e.g., Rudin, 1991), there exists a $L < \infty$ such that $\|\theta^*\|_\infty \leq L\|\theta^*\|$. In particular, we have $\|\theta^*\|_\infty < \infty$, from which we conclude by (5), that $w_m = 0$ holds for all m, which contradicts our assumption.

Hence, Prop. 12 can be applied,[1] which yields that (3) is equivalent to

$$\begin{aligned} \inf_{w,b,\theta} \quad & \tilde{C} \sum_{i=1}^{n} V\Big(\sum_{m=1}^{M} \langle w_m, \psi_m(x) \rangle + b,\, y_i \Big) + \frac{1}{2} \sum_{m=1}^{M} \frac{\|w_m\|_2^2}{\theta_m} \\ \text{s.t.} \quad & \|\theta\|^2 \leq \tau, \end{aligned}$$

for some $\tau > 0$. Consider the optimal solution $(w^\star, b^\star, \theta^\star)$ corresponding to a given parametrization $(\tilde{C}, \tau)$. For any $\lambda > 0$, the bijective transformation $(\tilde{C}, \tau) \mapsto (\lambda^{-1/2}\tilde{C}, \lambda\tau)$ will yield $(w^\star, b^\star, \lambda^{1/2}\theta^\star)$ as optimal solution. Applying the transformation with $\lambda := 1/\tau$ and setting $C = \tilde{C}\tau^{\frac{1}{2}}$ as well as $\kappa = \tau^{-1/2}$ yields Optimization Problem (4), which was to be shown. ■

Zien and Ong (2007) also show that the MKL optimization problems by Bach et al. (2004), Sonnenburg et al. (2006a), 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 (2007) lies in the same equivalence class as Bach et al. (2004), Sonnenburg et al. (2006a), Rakotomamonjy et al. (2007) and Zien and Ong (2007). In addition, our result shows the coupling between trade-off parameter C and the regularization parameter μ in Equation (3): tweaking one also changes the other and vice versa. Theorem 1 implies that optimizing C in Optimization Problem (4) implicitly searches the regularization path for the parameter μ of Equation (3). In the remainder, we will therefore focus on the formulation in Optimization Problem (4), as a single parameter is preferable in terms of model selection.

2.3 MKL in Dual Space

In this section we study the generalized MKL approach of the previous section in the dual space. Let us begin with rewriting Optimization Problem (4) by expanding the decision values into slack variables as follows

$$\begin{aligned} \inf_{w,b,t,\theta} \quad & C \sum_{i=1}^{n} V(t_i,\, y_i) + \frac{1}{2} \sum_{m=1}^{M} \frac{\|w_m\|_{\mathcal{H}_m}^2}{\theta_m} \\ \text{s.t.} \quad & \forall i: \ \sum_{m=1}^{M} \langle w_m, \psi_m(x_i) \rangle_{\mathcal{H}_m} + b = t_i \ ; \quad \|\theta\|^2 \leq 1 \ ; \quad \theta \geq \mathbf{0} \ , \end{aligned} \tag{6}$$

where $\|\cdot\|$ is an arbitrary norm in $\mathbb{R}^m$ and $\|\cdot\|_{\mathcal{H}_M}$ denotes the Hilbertian norm of $\mathcal{H}_m$. Applying Lagrange's theorem re-incorporates the constraints into the objective by introducing Lagrangian

1. Note that after a coordinate transformation, we can assume that $\mathcal{H}$ is finite dimensional (see Schölkopf et al., 1999).

multipliers $\alpha \in \mathbb{R}^n$, $\beta \in \mathbb{R}_+$, and $\gamma \in \mathbb{R}^M$. The Lagrangian saddle point problem is then given by

$$\sup_{\substack{\alpha,\beta,\gamma:\\ \beta\geq 0,\gamma\geq \mathbf{0}}} \inf_{w,b,t,\theta} \quad C\sum_{i=1}^{n} V(t_i,\, y_i) + \frac{1}{2}\sum_{m=1}^{M} \frac{\|w_m\|^2_{\mathcal{H}_m}}{\theta_m} - \sum_{i=1}^{n} \alpha_i \left(\sum_{m=1}^{M} \langle w_m, \psi_m(x_i)\rangle_{\mathcal{H}_m} + b - t_i \right) + \beta\left(\frac{1}{2}\|\theta\|^2 - \frac{1}{2}\right) - \gamma^\top \theta.$$

Denoting the Lagrangian by $\mathcal{L}$ and setting its first partial derivatives with respect to w and b to 0 reveals the optimality conditions

$$\mathbf{1}^\top \alpha = 0;$$
$$w_m = \theta_m \sum_{i=1}^{n} \alpha_i \psi_m(x_i), \;\; \forall\, m = 1,\ldots,M.$$

Resubstituting the above equations yields

$$\sup_{\substack{\alpha,\beta,\gamma:\ \mathbf{1}^\top\alpha=0,\\ \beta\geq 0,\gamma\geq \mathbf{0}}} \inf_{t,\theta} \quad C\sum_{i=1}^{n} (V(t_i,\, y_i) + \alpha_i t_i) - \frac{1}{2}\sum_{m=1}^{M} \theta_m \alpha^\top K_m \alpha + \beta\left(\frac{1}{2}\|\theta\|^2 - \frac{1}{2}\right) - \gamma^\top\theta,$$

which can also be written as

$$\sup_{\substack{\alpha,\beta,\gamma:\ \mathbf{1}^\top\alpha=0,\\ \beta\geq 0,\gamma\geq \mathbf{0}}} -C\sum_{i=1}^{n} \sup_{t_i}\left(-\frac{\alpha_i}{C}t_i - V(t_i,\, y_i)\right) - \beta \sup_{\theta}\left(\frac{1}{\beta}\sum_{m=1}^{M}\left(\frac{1}{2}\alpha^\top K_m\alpha + \gamma_m\right)\theta_m - \frac{1}{2}\|\theta\|^2\right) - \frac{1}{2}\beta.$$

As a consequence, we now may express the Lagrangian as[2]

$$\sup_{\alpha,\beta,\gamma:\ \mathbf{1}^\top\alpha=0,\beta\geq 0,\gamma\geq\mathbf{0}} -C\sum_{i=1}^{n} V^*\left(-\frac{\alpha_i}{C},\, y_i\right) - \frac{1}{\beta}\left\|\left(\frac{1}{2}\alpha^\top K_m\alpha + \gamma_m\right)_{m=1}^{M}\right\|_*^2 - \frac{1}{2}\beta, \tag{7}$$

where $h^*(x) = \sup_u x^\top u - h(u)$ denotes the Fenchel-Legendre conjugate of a function h and $\|\cdot\|_*$ denotes the *dual norm*, that is, the norm defined via the identity $\frac{1}{2}\|\cdot\|_*^2 := \left(\frac{1}{2}\|\cdot\|^2\right)^*$. In the following, we call V^* the *dual loss*. Equation (7) now has to be maximized with respect to the dual variables α, β, subject to $\mathbf{1}^\top\alpha = 0$ and $\beta \geq 0$. Let us ignore for a moment the non-negativity constraint on β and solve $\partial\mathcal{L}/\partial\beta = 0$ for the unbounded β. Setting the partial derivative to zero allows to express the optimal β as

$$\beta = \left\|\left(\frac{1}{2}\alpha^\top K_m\alpha + \gamma_m\right)_{m=1}^{M}\right\|_*. \tag{8}$$

Obviously, at optimality, we always have $\beta \geq 0$. We thus discard the corresponding constraint from the optimization problem and plugging Equation (8) into Equation (7) results in the following *dual* optimization problem:

2. We employ the notation $s = (s_1,\ldots,s_M)^\top = (s_m)_{m=1}^M$ for $s \in \mathbb{R}^M$.

2.3.1 GENERAL DUAL MKL OPTIMIZATION PROBLEM

$$\sup_{\alpha,\gamma:\ \mathbf{1}^\top\alpha=0,\gamma\geq\mathbf{0}} \quad -C\sum_{i=1}^{n} V^*\left(-\frac{\alpha_i}{C},\, y_i\right) - \left\|\left(\frac{1}{2}\alpha^\top K_m\alpha+\gamma_m\right)_{m=1}^{M}\right\|_* . \tag{9}$$

The above dual generalizes multiple kernel learning to arbitrary convex loss functions and norms.[3] Note that for the most common choices of norms (for example, ℓ_p-norm, weighted ℓ_p-norms, and sum of ℓ_p-norms; but not the norms discussed in Section 3.5) it holds $\gamma^* = 0$ in the optimal point so that the γ-term can be discarded and the above reduces to an optimization problem that solely depends on α. Also note that if the loss function is continuous (e.g., hinge loss), the supremum is also a maximum. The threshold b can be recovered from the solution by applying the KKT conditions.

The above dual can be characterized as follows. We start by noting that the expression in Optimization Problem (9) is a composition of two terms, first, the left hand side term, which depends on the conjugate loss function V^*, and, second, the right hand side term which depends on the conjugate norm. The right hand side can be interpreted as a regularizer on the quadratic terms that, according to the chosen norm, smoothens the solutions. Hence we have a decomposition of the dual into a loss term (in terms of the dual loss) and a regularizer (in terms of the dual norm). For a specific choice of a pair $(V, \|\cdot\|)$ we can immediately recover the corresponding dual by computing the pair of conjugates $(V^*, \|\cdot\|_*)$ (for a comprehensive list of dual losses see Rifkin and Lippert, 2007, Table 3). In the next section, this is illustrated by means of well-known loss functions and regularizers.

At this point we would like to highlight some properties of Optimization Problem (9) that arise due to our dualization technique. While approaches that firstly apply the representer theorem and secondly optimize in the primal such as Chapelle (2006) also can employ general loss functions, the resulting loss terms depend on all optimization variables. By contrast, in our formulation the dual loss terms are of a much simpler structure and they only depend on a single optimization variable α_i. A similar dualization technique yielding singly-valued dual loss terms is presented in Rifkin and Lippert (2007); it is based on Fenchel duality and limited to strictly positive definite kernel matrices. Our technique, which uses Lagrangian duality, extends the latter by allowing for positive semi-definite kernel matrices.

3. Recovering Previous MKL Formulations as Special Instances

In this section we show that existing MKL-based learners are subsumed by the generalized formulation in Optimization Problem (9). It is helpful for what is coming up to note that for most (but not all; see Section 3.5) choices of norms it holds $\gamma^* = 0$ in the generalized dual MKL problem (9), so that it simplifies to:

$$\sup_{\alpha:\ \mathbf{1}^\top\alpha=0} \quad -C\sum_{i=1}^{n} V^*\left(-\frac{\alpha_i}{C},\, y_i\right) - \frac{1}{2}\left\|\left(\alpha^\top K_m\alpha\right)_{m=1}^{M}\right\|_* . \tag{10}$$

3. We can even employ non-convex losses and still the dual will be a convex problem; however, it might suffer from a duality gap.

3.1 Support Vector Machines with Unweighted-Sum Kernels

First, we note that the support vector machine with an unweighted-sum kernel can be recovered as a special case of our model. To see this, we consider the regularized risk minimization problem using the hinge loss function $V(t,y) = \max(0, 1-ty)$ and the regularizer $\|\theta\|_\infty$. We then can obtain the corresponding dual in terms of Fenchel-Legendre conjugate functions as follows.

We first note that the dual loss of the hinge loss is $V^*(t,y) = \frac{t}{y}$ if $-1 \leq \frac{t}{y} \leq 0$ and ∞ elsewise (Rifkin and Lippert, 2007, Table 3). Hence, for each i the term $V^*\left(-\frac{\alpha_i}{C}, y_i\right)$ of the generalized dual, that is, Optimization Problem (9), translates to $-\frac{\alpha_i}{Cy_i}$, provided that $0 \leq \frac{\alpha_i}{y_i} \leq C$. Employing a variable substitution of the form $\alpha_i^{\text{new}} = \frac{\alpha_i}{y_i}$, Optimization Problem (9) translates to

$$\max_{\alpha,\gamma:\ \gamma \geq 0} \quad \mathbf{1}^\top \alpha - \left\| \left(\frac{1}{2} \alpha^\top Y K_m Y \alpha + \gamma_m \right)_{m=1}^{M} \right\|_*, \quad \text{s.t.} \quad y^\top \alpha = 0 \ \text{ and } \ \mathbf{0} \leq \alpha \leq C\mathbf{1}, \tag{11}$$

where we denote $Y = \operatorname{diag}(y)$. The primal ℓ_∞-norm penalty $\|\theta\|_\infty$ is dual to $\|\theta\|_1$, hence, via the identity $\|\cdot\|_* = \|\cdot\|_1$ the right hand side of the last equation translates to $\sum_{m=1}^M \alpha^\top Y K_m Y \alpha$, and we note that $\gamma^* = 0$ in the optimal point. Combined with (11) this leads to the dual

$$\max_{\alpha} \quad \mathbf{1}^\top \alpha - \sum_{m=1}^{M} \alpha^\top Y K_m Y \alpha, \quad \text{s.t.} \quad y^\top \alpha = 0 \ \text{ and } \ \mathbf{0} \leq \alpha \leq C\mathbf{1},$$

which is precisely an SVM with an unweighted-sum kernel.

3.2 QCQP MKL of Lanckriet et al. (2004)

A common approach in multiple kernel learning is to employ regularizers of the form

$$\Omega(\theta) = \|\theta\|_1. \tag{12}$$

This so-called ℓ_1-norm regularizers are specific instances of *sparsity-inducing* regularizers. The obtained kernel mixtures usually have a considerably large fraction of zero entries, and hence equip the MKL problem by the favor of interpretable solutions. Sparse MKL is a special case of our framework; to see this, note that the conjugate of (12) is $\|\cdot\|_\infty$. Recalling the definition of an ℓ_p-norm, the right hand side of Optimization Problem (9) translates to $\max_{m\in\{1,\ldots,M\}} \alpha^\top Y K_m Y \alpha$. The maximum can subsequently be expanded into a slack variable ξ, resulting in

$$\begin{aligned} &\sup_{\alpha,\xi} \quad \mathbf{1}^\top \alpha - \xi \\ &\text{s.t.} \quad \forall\, m: \ \frac{1}{2} \alpha^\top Y K_m Y \alpha \leq \xi\,; \quad y^\top \alpha = 0\,; \quad \mathbf{0} \leq \alpha \leq C\mathbf{1}, \end{aligned}$$

which is the original QCQP formulation of MKL, firstly given by Lanckriet et al. (2004).

3.3 A Smooth Variant of Group Lasso

Yuan and Lin (2006) studied the following optimization problem for the special case $\mathcal{H}_m = \mathbb{R}^{d_m}$ and $\psi_m = \mathrm{id}_{\mathbb{R}^{d_m}}$, also known as group lasso,

$$\min_{w} \quad \frac{C}{2} \sum_{i=1}^{n} \left(y_i - \sum_{m=1}^{M} \langle w_m, \psi_m(x_i) \rangle_{\mathcal{H}_m} \right)^2 + \frac{1}{2} \sum_{m=1}^{M} \|w_m\|_{\mathcal{H}_m}. \tag{13}$$

The above problem has been solved by active set methods in the primal (Roth and Fischer, 2008). We sketch an alternative approach based on dual optimization. First, we note that Equation (13) can be equivalently expressed as (Micchelli and Pontil, 2005, Lemma 26)

$$\inf_{w,\theta:\theta\geq\mathbf{0}} \quad \frac{C}{2}\sum_{i=1}^{n}\left(y_i - \sum_{m=1}^{M}\langle w_m, \psi_m(x_i)\rangle_{\mathcal{H}_m}\right)^2 + \frac{1}{2}\sum_{m=1}^{M}\frac{\|w_m\|_{\mathcal{H}_m}^2}{\theta_m}, \quad \text{s.t.} \quad \|\theta\|_1^2 \leq 1.$$

The dual of $V(t,y) = \frac{1}{2}(y-t)^2$ is $V^*(t,y) = \frac{1}{2}t^2 + ty$ and thus the corresponding group lasso dual can be written as

$$\max_{\alpha} \quad y^\top\alpha - \frac{1}{2C}\|\alpha\|_2^2 - \frac{1}{2}\left\|\left(\alpha^\top Y K_m Y \alpha\right)_{m=1}^{M}\right\|_\infty,$$

which can be expanded into the following QCQP

$$\begin{aligned}\sup_{\alpha,\xi} \quad & y^\top\alpha - \frac{1}{2C}\|\alpha\|_2^2 - \xi \\ \text{s.t.} \quad \forall m: \quad & \frac{1}{2}\alpha^\top Y K_m Y \alpha \leq \xi.\end{aligned}$$

For small n, the latter formulation can be handled efficiently by QCQP solvers. However, the quadratic constraints caused by the non-smooth ℓ_∞-norm in the objective still are computationally too demanding. As a remedy, we propose the following unconstrained variant based on ℓ_p-norms ($1 < p < \infty$), given by

$$\max_{\alpha} \quad y^\top\alpha - \frac{1}{2C}\|\alpha\|_2^2 - \frac{1}{2}\left\|\left(\alpha^\top Y K_m Y \alpha\right)_{m=1}^{M}\right\|_{p^*}.$$

It is straightforward to verify that the above objective function is differentiable in any $\alpha \in \mathbb{R}^n$ (in particular, notice that the ℓ_p-norm function is differentiable for $1 < p < \infty$) and hence the above optimization problem can be solved very efficiently by, for example, limited memory quasi-Newton descent methods (Liu and Nocedal, 1989).

3.4 Density Level-Set Estimation

Density level-set estimators are frequently used for anomaly/novelty detection tasks (Markou and Singh, 2003a,b). Kernel approaches, such as one-class SVMs (Schölkopf et al., 2001) and Support Vector Domain Descriptions (Tax and Duin, 1999) can be cast into our MKL framework by employing loss functions of the form $V(t) = \max(0, 1-t)$. This gives rise to the primal

$$\inf_{w,\theta:\theta\geq\mathbf{0}} \quad C\sum_{i=1}^{n}\max\left(0, \sum_{m=1}^{M}\langle w_m, \psi_m(x_i)\rangle_{\mathcal{H}_m}\right) + \frac{1}{2}\sum_{m=1}^{M}\frac{\|w_m\|_{\mathcal{H}_m}^2}{\theta_m}, \quad \text{s.t.} \quad \|\theta\|^2 \leq 1.$$

Noting that the dual loss is $V^*(t) = t$ if $-1 \leq t \leq 0$ and ∞ elsewise, we obtain the following generalized dual

$$\sup_{\alpha} \quad \mathbf{1}^\top\alpha - \frac{1}{2}\left\|\left(\alpha^\top K_m \alpha\right)_{m=1}^{M}\right\|_{p^*}, \quad \text{s.t.} \quad \mathbf{0} \leq \alpha \leq C\mathbf{1},$$

which has been studied by Sonnenburg et al. (2006a) and Rakotomamonjy et al. (2008) for ℓ_1-norm, and by Kloft et al. (2009b) for ℓ_p-norms.

3.5 Non-Isotropic Norms

In practice, it is often desirable for an expert to incorporate prior knowledge about the problem domain. For instance, an expert could provide estimates of the interactions of kernels $\{K_1,...,K_M\}$ in the form of an $M \times M$ matrix E. Alternatively, E could be obtained by computing pairwise kernel alignments $E_{ij} = \frac{<K_i,K_j>}{\|K_i\| \, \|K_j\|}$ given a dot product on the space of kernels such as the Frobenius dot product (Ong et al., 2005). In a third scenario, E could be a diagonal matrix encoding the a priori importance of kernels—it might be known from pilot studies that a subset of the employed kernels is inferior to the remaining ones.

All those scenarios can be handled within our framework by considering non-isotropic regularizers of the form[4]

$$\|\theta\|_{E^{-1}} = \sqrt{\theta^\top E^{-1}\theta} \;\text{ with }\; E \succ 0,$$

where E^{-1} is the matrix inverse of E.

However, this choice of a norm is quite different from what we have seen before: let us consider Optimization Problem (9); for non-isotropic norms we in general do not have $\gamma^* = 0$ in the optimal point so that this OP does not simplify to the dual (10) as in the subsections before. Instead we have to work with (9) directly. To this end, note that for the dual norm it holds $\left(\frac{1}{2}\|\cdot\|^2_{E^{-1}}\right)^* = \frac{1}{2}\|\cdot\|^2_E$, so that we obtain from (9) the following dual

$$\sup_{\alpha,\gamma:\, \mathbf{1}^\top\alpha=\mathbf{0},\gamma\geq\mathbf{0}} \quad -C\sum_{i=1}^n V^*\left(-\frac{\alpha_i}{C},\, y_i\right) - \left\|\left(\frac{1}{2}\alpha^\top K_m\alpha+\gamma_m\right)_{m=1}^M\right\|_E,$$

which is the desired non-isotropic MKL problem.

4. ℓ_p-Norm Multiple Kernel Learning

In this work, we propose to use non-sparse and thus more robust kernel mixtures by employing an ℓ_p-norm constraint with $p > 1$, rather than the traditionally used ℓ_1-norm constraint, on the mixing coefficients (Kloft et al., 2009a). To this end, we employ non-sparse norms of the form $\|\theta\|_p = (\sum_{m=1}^M \theta_m^p)^{1/p}$, $1 < p < \infty$.[5] From the unifying framework of Section 2 we obtain the following ℓ_p-norm MKL primal:

4.1 Primal ℓ_p-norm MKL Optimization Problem

$$\begin{aligned} \inf_{w,b,\theta:\theta\geq\mathbf{0}} \quad & C\sum_{i=1}^n V\left(\sum_{m=1}^M \langle w_m, \psi_m(x_i)\rangle_{\mathcal{H}_m} + b,\, y_i\right) + \frac{1}{2}\sum_{m=1}^M \frac{\|w_m\|^2_{\mathcal{H}_m}}{\theta_m} \\ \text{s.t.} \quad & \|\theta\|_p^2 \leq 1. \end{aligned} \tag{14}$$

Using that the dual norm of the ℓ_p-norm is the ℓ_{p^*}-norm, where $p^* := \frac{p}{p-1}$, and noting that $\gamma^* = 0$ in the optimal point, we obtain from Optimization Problem (9) the following ℓ_p-norm MKL dual:

4. This idea is inspired by the Mahalanobis distance (Mahalanobis, 1936).
5. While the upcoming reasoning also holds for weighted ℓ_p-norms, the extension to more general norms, such as the ones described in Section 3.5, is left for future work.

4.2 Dual ℓ_p-norm MKL Optimization Problem

$$\sup_{\alpha:\mathbf{1}^\top\alpha=\mathbf{0}} \quad -C\sum_{i=1}^{n} V^*\left(-\frac{\alpha_i}{C}, y_i\right) - \frac{1}{2}\left\|\left(\alpha^\top K_m \alpha\right)_{m=1}^{M}\right\|_{p^*}.$$

In the special case of hinge loss minimization, we obtain the optimization problem

$$\sup_{\alpha} \quad \mathbf{1}^\top\alpha - \frac{1}{2}\left\|\left(\alpha^\top Y K_m Y \alpha\right)_{m=1}^{M}\right\|_{p^*}, \quad \text{s.t.} \quad y^\top\alpha = 0 \text{ and } \mathbf{0} \le \alpha \le C\mathbf{1}. \tag{15}$$

In the subsequent sections, we will propose an efficient optimization algorithm for Optimization Problem (15) (Section 4.3) and proof its convergence (Section 4.3.3). Later we derive generalization bounds (Section 5), and analyze ℓ_p-norm MKL empirically using artificial and real-world data sets (Section 6).

4.3 Optimization Strategies

The dual as given in Optimization Problem (15) does not lend itself to efficient large-scale optimization in a straight-forward fashion, for instance by direct application of standard approaches like gradient descent. Instead, it is beneficial to exploit the structure of the MKL cost function by alternating between optimizing w.r.t. the mixings θ and w.r.t. the remaining variables. Most recent MKL solvers (e.g., Rakotomamonjy et al., 2008; Xu et al., 2009; Nath et al., 2009) do so by setting up a two-layer optimization procedure: a master problem, which is parameterized only by θ, is solved to determine the kernel mixture; to solve this master problem, repeatedly a slave problem is solved which amounts to training a standard SVM on a mixture kernel. Importantly, for the slave problem, the mixture coefficients are fixed, such that conventional, efficient SVM optimizers can be recycled. Consequently these two-layer procedures are commonly implemented as *wrapper* approaches. Albeit appearing advantageous, wrapper methods suffer from two shortcomings: (i) Due to kernel cache limitations, the kernel matrices have to be pre-computed and stored or many kernel computations have to be carried out repeatedly, inducing heavy wastage of either memory or time. (ii) The slave problem is always optimized to the end (and many convergence proofs seem to require this), although most of the computational time is spend on the non-optimal mixtures. Certainly suboptimal slave solutions would already suffice to improve far-from-optimal θ in the master problem.

Due to these problems, MKL is prohibitive when learning with a multitude of kernels and on large-scale data sets as commonly encountered in many data-intense real world applications such as bioinformatics, web mining, databases, and computer security. The optimization approach presented in this paper decomposes the MKL problem into smaller subproblems (Platt, 1999; Joachims, 1999; Fan et al., 2005) by establishing a wrapper-like scheme *within* the decomposition algorithm.

Our algorithm is embedded into the large-scale framework of Sonnenburg et al. (2006a) and extends it to the optimization of non-sparse kernel mixtures induced by an ℓ_p-norm penalty. Our strategy alternates between minimizing the primal problem (6) w.r.t. θ via a simple analytical update formula and with incomplete optimization w.r.t. all other variables which, however, is performed in terms of the dual variables α. Optimization w.r.t. α is performed by chunking optimizations with minor iterations. Convergence of our algorithm is proven under typical technical regularity assumptions.

4.3.1 A SIMPLE WRAPPER APPROACH BASED ON AN ANALYTICAL UPDATE

We first present an easy-to-implement wrapper version of our optimization approach to multiple kernel learning. The interleaved decomposition algorithm is deferred to the next section.

To derive the new algorithm, we divide the optimization variables of the primal problem (14) into two groups, (w,b) on one hand and θ on the other. Our algorithm will alternatingly operate on those two groups via a block coordinate descent algorithm, also known as the *non-linear block Gauss-Seidel method*. Thereby the optimization w.r.t. θ will be carried out analytically and the (w,b)-step will be computed in the dual, if needed.

The basic idea of our first approach is that for a given, fixed set of primal variables (w,b), the optimal θ in the primal problem (14) can be calculated analytically as the following proposition shows.

Proposition 2 *Let V be a convex loss function, be $p>1$. Given fixed (possibly suboptimal) $w\neq\mathbf{0}$ and b, the minimal θ in Optimization Problem* (14) *is attained for*

$$\theta_m = \frac{\|w_m\|_{\mathcal{H}_m}^{\frac{2}{p+1}}}{\left(\sum_{m'=1}^{M}\|w_{m'}\|_{\mathcal{H}_{m'}}^{\frac{2p}{p+1}}\right)^{1/p}}, \qquad \forall m=1,\ldots,M. \tag{16}$$

Proof [6] We start the derivation, by equivalently translating Optimization Problem (14) via Theorem 1 into

$$\inf_{w,b,\theta:\theta\geq\mathbf{0}} \quad \tilde{C}\sum_{i=1}^{n} V\left(\sum_{m=1}^{M}\langle w_m,\psi_m(x_i)\rangle_{\mathcal{H}_m}+b,\, y_i\right)+\frac{1}{2}\sum_{m=1}^{M}\frac{\|w_m\|_{\mathcal{H}_m}^2}{\theta_m}+\frac{\mu}{2}\|\theta\|_p^2, \tag{17}$$

with $\mu>0$. Suppose we are given fixed (w,b), then setting the partial derivatives of the above objective w.r.t. θ to zero yields the following condition on the optimality of θ,

$$-\frac{\|w_m\|_{\mathcal{H}_m}^2}{2\theta_m^2}+\mu\cdot\frac{\partial\left(\frac{1}{2}\|\theta\|_p^2\right)}{\partial\theta_m}=0, \quad \forall m=1,\ldots,M. \tag{18}$$

The first derivative of the ℓ_p-norm with respect to the mixing coefficients can be expressed as

$$\frac{\partial\left(\frac{1}{2}\|\theta\|_p^2\right)}{\partial\theta_m}=\theta_m^{p-1}\|\theta\|_p^{2-p},$$

and hence Equation (18) translates into the following optimality condition,

$$\exists\zeta \quad \forall m=1,\ldots,M: \qquad \theta_m=\zeta\|w_m\|_{\mathcal{H}_m}^{\frac{2}{p+1}}. \tag{19}$$

Because $w\neq 0$, using the same argument as in the proof of Theorem 1, the constraint $\|\theta\|_p^2\leq 1$ in (17) is at the upper bound, that is, $\|\theta\|_p=1$ holds for an optimal θ. Inserting (19) in the latter equation leads to $\zeta=\left(\sum_{m=1}^{M}\|w_m\|_{\mathcal{H}_m}^{2p/p+1}\right)^{1/p}$. Resubstitution into (19) yields the claimed formula

6. We remark that a more general result can be obtained by an alternative proof using Hölder's inequality (see Lemma 26 in Micchelli and Pontil, 2005).

(16). ■

Second, we consider how to optimize Optimization Problem (14) w.r.t. the remaining variables (w,b) for a given set of mixing coefficients θ. Since optimization often is considerably easier in the dual space, we fix θ and build the partial Lagrangian of Optimization Problem (14) w.r.t. all other primal variables w, b. The resulting dual problem is of the form (detailed derivations omitted)

$$\sup_{\alpha:\mathbf{1}^\top\alpha=0} -C\sum_{i=1}^{n} V^*\left(-\frac{\alpha_i}{C},\, y_i\right) - \frac{1}{2}\sum_{m=1}^{M}\theta_m\alpha^\top K_m\alpha, \tag{20}$$

and the KKT conditions yield $w_m = \theta_m\sum_{i=1}^{n}\alpha_i\psi_m(x_i)$ in the optimal point, hence

$$\|w_m\|^2 = \theta_m^2\alpha K_m\alpha, \quad \forall\, m=1,\ldots,M. \tag{21}$$

We now have all ingredients (i.e., Equations (16), (20)–(21)) to formulate a simple macro-wrapper algorithm for ℓ_p-norm MKL training:

Algorithm 1 Simple $\ell_{p>1}$-norm MKL wrapper-based training algorithm. The analytical updates of θ and the SVM computations are optimized alternatingly.

1: **input:** feasible α and θ
2: **while** optimality conditions are not satisfied **do**
3: Compute α according to Equation (20) (e.g., SVM)
4: Compute $\|w_m\|^2$ for all $m=1,\ldots,M$ according to Equation (21)
5: Update θ according to Equation (16)
6: **end while**

The above algorithm alternatingly solves a convex risk minimization machine (e.g., SVM) w.r.t. the actual mixture θ (Equation (20)) and subsequently computes the analytical update according to Equation (16) and (21). It can, for example, be stopped based on changes of the objective function or the duality gap within subsequent iterations.

4.3.2 TOWARDS LARGE-SCALE MKL—INTERLEAVING SVM AND MKL OPTIMIZATION

However, a disadvantage of the above wrapper approach still is that it deploys a full blown kernel matrix. We thus propose to interleave the SVM optimization of SVMlight with the θ- and α-steps at training time. We have implemented this so-called *interleaved* algorithm in Shogun for hinge loss, thereby promoting sparse solutions in α. This allows us to solely operate on a small number of active variables.[7] The resulting interleaved optimization method is shown in Algorithm 2. Lines 3-5 are standard in chunking based SVM solvers and carried out by SVM$^{\text{light}}$ (note that Q is chosen as described in Joachims, 1999). Lines 6-7 compute SVM-objective values. Finally, the analytical θ-step is carried out in Line 9. The algorithm terminates if the maximal KKT violation (cf. Joachims, 1999) falls below a predetermined precision ε and if the normalized maximal constraint violation $|1-\frac{\omega}{\omega_{old}}|<\varepsilon_{mkl}$ for the MKL-step, where ω denotes the MKL objective function value (Line 8).

7. In practice, it turns out that the kernel matrix of active variables typically is about of the size 40×40, even when we deal with ten-thousands of examples.

Algorithm 2 ℓ_p-Norm MKL chunking-based training algorithm via analytical update. Kernel weighting θ and (signed) SVM α are optimized interleavingly. The accuracy parameter ε and the subproblem size Q are assumed to be given to the algorithm.

Initialize: $g_{m,i} = \hat{g}_i = \alpha_i = 0, \forall i = 1,...,n; \quad L = S = -\infty; \quad \theta_m = \sqrt[p]{1/M}, \forall m = 1,...,M$
iterate
  Select Q variables $\alpha_{i_1}, \ldots, \alpha_{i_Q}$ based on the gradient $\hat{\mathbf{g}}$ of (20) w.r.t. α
  Store $\alpha^{old} = \alpha$ and then update α according to (20) with respect to the selected variables
  Update gradient $g_{m,i} \leftarrow g_{m,i} + \sum_{q=1}^{Q} (\alpha_{i_q} - \alpha_{i_q}^{old}) k_m(x_{i_q}, x_i), \ \forall\, m = 1, \ldots, M,\ i = 1, \ldots, n$
  Compute the quadratic terms $S_m = \frac{1}{2} \sum_i g_{m,i} \alpha_i, \ q_m = 2\theta_m^2 S_m, \ \forall m = 1, \ldots, M$
  $L_{old} = L, \quad L = \sum_i y_i \alpha_i, \quad S_{old} = S, \quad S = \sum_m \theta_m S_m$
  if $|1 - \frac{L-S}{L_{old} - S_{old}}| \geq \varepsilon$
    $\theta_m = (q_m)^{1/(p+1)} \Big/ \left(\sum_{m'=1}^{M} (q_{m'})^{p/(p+1)} \right)^{1/p}, \quad \forall\, m = 1, \ldots, M$
  else
    break
  end if
  $\hat{g}_i = \sum_m \theta_m g_{m,i}$ for all $i = 1, \ldots, n$

4.3.3 CONVERGENCE PROOF FOR $p > 1$

In the following, we exploit the primal view of the above algorithm as a nonlinear block Gauss-Seidel method, to prove convergence of our algorithms. We first need the following useful result about convergence of the nonlinear block Gauss-Seidel method in general.

Proposition 3 (Bertsekas, 1999, Prop. 2.7.1) *Let $\mathcal{X} = \bigotimes_{m=1}^{M} \mathcal{X}_m$ be the Cartesian product of closed convex sets $\mathcal{X}_m \subset \mathbb{R}^{d_m}$, be $f : \mathcal{X} \to \mathbb{R}$ a continuously differentiable function. Define the nonlinear block Gauss-Seidel method recursively by letting $x^0 \in \mathcal{X}$ be any feasible point, and be*

$$x_m^{k+1} = \operatorname*{argmin}_{\xi \in \mathcal{X}_m} f\left(x_1^{k+1}, \cdots, x_{m-1}^{k+1}, \xi, x_{m+1}^{k}, \cdots, x_M^{k}\right), \ \forall m = 1, \ldots, M. \tag{22}$$

Suppose that for each m and $x \in \mathcal{X}$, the minimum

$$\min_{\xi \in \mathcal{X}_m} f\left(x_1, \cdots, x_{m-1}, \xi, x_{m+1}, \cdots, x_M\right)$$

is uniquely attained. Then every limit point of the sequence $\{x^k\}_{k \in \mathbb{N}}$ is a stationary point.

The proof can be found in Bertsekas (1999), p. 268-269. The next proposition basically establishes convergence of the proposed ℓ_p-norm MKL training algorithm.

Theorem 4 *Let V be the hinge loss and be $p > 1$. Let the kernel matrices $K_1, \ldots, K_M$ be positive definite. Then every limit point of Algorithm 1 is a globally optimal point of Optimization Problem* (14)*. Moreover, suppose that the SVM computation is solved exactly in each iteration, then the same holds true for Algorithm 2.*

Proof If we ignore the numerical speed-ups, then the Algorithms 1 and 2 coincide for the hinge loss. Hence, it suffices to show the wrapper algorithm converges.

To this aim, we have to transform Optimization Problem (14) into a form such that the requirements for application of Prop. 3 are fulfilled. We start by expanding Optimization Problem (14) into

$$\begin{aligned}\min_{w,b,\xi,\theta} \quad & C\sum_{i=1}^{n}\xi_i + \frac{1}{2}\sum_{m=1}^{M}\frac{\|w_m\|^2_{\mathcal{H}_m}}{\theta_m},\\ \text{s.t.} \quad & \forall i: \sum_{m=1}^{M}\langle w_m, \psi_m(x_i)\rangle_{\mathcal{H}_m} + b \geq 1-\xi_i; \quad \xi \geq 0; \quad \|\theta\|_p^2 \leq 1; \quad \theta \geq \mathbf{0},\end{aligned}$$

thereby extending the second block of variables, (w,b), into (w,b,ξ). Moreover, we note that after an application of the representer theorem[8] (Kimeldorf and Wahba, 1971) we may without loss of generality assume $\mathcal{H}_m = \mathbb{R}^n$.

In the problem's current form, the possibility of an optimal $\theta_m = 0$ while $w_m \neq 0$ renders the objective function nondifferentiable. This hinders the application of Prop. 3. Fortunately, it follows from Prop. 2 (note that $K_m \succ 0$ implies $w \neq \mathbf{0}$) that this case is impossible for $p > 1$. We therefore can substitute the constraint $\theta \geq \mathbf{0}$ by $\theta > \mathbf{0}$ for all m without changing the optimum. In order to maintain the closeness of the feasible set we subsequently apply a bijective coordinate transformation $\phi : \mathbb{R}^M_+ \to \mathbb{R}^M$ with $\theta_m^{\text{new}} = \phi_m(\theta_m) = \log(\theta_m)$, resulting in the following equivalent problem,

$$\begin{aligned}\inf_{w,b,\xi,\theta} \quad & C\sum_{i=1}^{n}\xi_i + \frac{1}{2}\sum_{m=1}^{M}\exp(-\theta_m)\|w_m\|^2_{\mathbb{R}^n},\\ \text{s.t.} \quad & \forall i: \sum_{m=1}^{M}\langle w_m, \psi_m(x_i)\rangle_{\mathbb{R}^n} + b \geq 1-\xi_i; \quad \xi \geq 0; \quad \|\exp(\theta)\|_p^2 \leq 1,\end{aligned}$$

where we employ the notation $\exp(\theta) = (\exp(\theta_1), \cdots, \exp(\theta_M))^\top$.

Applying the Gauss-Seidel method in Equation (22) to the base problem Optimization Problem (14) and to the reparametrized problem yields the same sequence of solutions $\{(w,b,\theta)^k\}_{k\in\mathbb{N}_0}$. The above problem now allows to apply Prop. 3 for the two blocks of coordinates $\theta \in \mathcal{X}_1$ and $(w,b,\xi) \in \mathcal{X}_2$: the objective is continuously differentiable and the sets $\mathcal{X}_1$ and $\mathcal{X}_2$ are closed and convex. To see the latter, note that $\|\cdot\|_p^2 \circ \exp$ is a convex function (cf., Section 3.2.4 in Boyd and Vandenberghe, 2004). Moreover, the minima in Equation (22) are uniquely attained: the (w,b)-step amounts to solving an SVM on a positive definite kernel mixture, and the analytical θ-step clearly yields unique solutions as well.

Hence, we conclude that every limit point of the sequence $\{(w,b,\theta)^k\}_{k\in\mathbb{N}}$ is a stationary point of Optimization Problem (14). For a convex problem, this is equivalent to such a limit point being globally optimal. ■

In practice, we are facing two problems. First, the standard Hilbert space setup necessarily implies that $\|w_m\| \geq 0$ for all m. However in practice this assumption may often be violated, either due to numerical imprecision or because of using an indefinite "kernel" function. However, for any $\|w_m\| \leq 0$ it also follows that $\theta_m^\star = 0$ as long as at least one strictly positive $\|w_{m'}\| > 0$ exists. This is because for any $\lambda < 0$ we have $\lim_{h\to 0, h>0} \frac{\lambda}{h} = -\infty$. Thus, for any m with $\|w_m\| \leq 0$, we

8. Note that the coordinate transformation into $\mathbb{R}^n$ can be explicitly given in terms of the empirical kernel map (Schölkopf et al., 1999).

can immediately set the corresponding mixing coefficients $\theta_m^\star$ to zero. The remaining θ are then computed according to Equation (2), and convergence will be achieved as long as at least one strictly positive $\|w_{m'}\| > 0$ exists in each iteration.

Second, in practice, the SVM problem will only be solved with finite precision, which may lead to convergence problems. Moreover, we actually want to improve the α only a little bit before recomputing θ since computing a high precision solution can be wasteful, as indicated by the superior performance of the interleaved algorithms (cf. Sect. 6.5). This helps to avoid spending a lot of α-optimization (SVM training) on a suboptimal mixture θ. Fortunately, we can overcome the potential convergence problem by ensuring that the primal objective decreases within each α-step. This is enforced in practice, by computing the SVM by a higher precision if needed. However, in our computational experiments we find that this precaution is not even necessary: even without it, the algorithm converges in all cases that we tried (cf. Section 6).

Finally, we would like to point out that the proposed block coordinate descent approach lends itself more naturally to combination with primal SVM optimizers like Chapelle (2006), LibLinear (Fan et al., 2008) or Ocas (Franc and Sonnenburg, 2008). Especially for linear kernels this is extremely appealing.

4.4 Technical Considerations

In this section we report on implementation details and discuss kernel normalization.

4.4.1 IMPLEMENTATION DETAILS

We have implemented the analytic optimization algorithm described in the previous Section, as well as the cutting plane and Newton algorithms by Kloft et al. (2009a), within the SHOGUN toolbox (Sonnenburg et al., 2010) for regression, one-class classification, and two-class classification tasks. In addition one can choose the optimization scheme, that is, decide whether the interleaved optimization algorithm or the wrapper algorithm should be applied. In all approaches any of the SVMs contained in SHOGUN can be used. Our implementation can be downloaded from `http://www.shogun-toolbox.org`.

In the more conventional family of approaches, the *wrapper algorithms*, an optimization scheme on θ wraps around a single kernel SVM. Effectively this results in alternatingly solving for α and θ. For the outer optimization (i.e., that on θ) SHOGUN offers the three choices listed above. The semi-infinite program (SIP) uses a traditional SVM to generate new violated constraints and thus requires a single kernel SVM. A linear program (for $p = 1$) or a sequence of quadratically constrained linear programs (for $p > 1$) is solved via GLPK[9] or IBM ILOG CPLEX[10]. Alternatively, either an analytic or a Newton update (for ℓ_p norms with $p > 1$) step can be performed, obviating the need for an additional mathematical programming software.

The second, much faster approach performs interleaved optimization and thus requires modification of the core SVM optimization algorithm. It is currently integrated into the chunking-based SVRlight and SVMlight. To reduce the implementation effort, we implement a single function `perform_mkl_step`$(\sum_\alpha,\ \text{obj}_m)$, that has the arguments $\sum_\alpha := \sum_{i=1}^n \alpha_i$ and $\text{obj}_m = \frac{1}{2}\alpha^T K_m \alpha$, that is, the current linear α-term and the SVM objectives for each kernel. This function is either, in the

9. GLPK can be found at `http://www.gnu.org/software/glpk/`.
10. ILOG CPLEX can be found at `http://www.ibm.com/software/integration/optimization/cplex/`.

interleaved optimization case, called as a callback function (after each chunking step or a couple of SMO steps), or it is called by the wrapper algorithm (after each SVM optimization to full precision).

Recovering Regression and One-Class Classification. It should be noted that one-class classification is trivially implemented using $\sum_\alpha = 0$ while support vector regression (SVR) is typically performed by internally translating the SVR problem into a standard SVM classification problem with twice the number of examples once positively and once negatively labeled with corresponding α and α^*. Thus one needs direct access to α^* and computes $\sum_\alpha = -\sum_{i=1}^n (\alpha_i + \alpha_i^*)\varepsilon - \sum_{i=1}^n (\alpha_i - \alpha_i^*) y_i$ (cf. Sonnenburg et al., 2006a). Since this requires modification of the core SVM solver we implemented SVR only for interleaved optimization and SVMlight.

Efficiency Considerations and Kernel Caching. Note that the choice of the size of the kernel cache becomes crucial when applying MKL to large scale learning applications.[11] While for the wrapper algorithms only a *single* kernel SVM needs to be solved and thus a single large kernel cache should be used, the story is different for interleaved optimization. Since one must keep track of the several partial MKL objectives obj_m, requiring access to individual kernel rows, the same cache size should be used for all sub-kernels.

4.4.2 KERNEL NORMALIZATION

The normalization of kernels is as important for MKL as the normalization of features is for training regularized linear or single-kernel models. This is owed to the bias introduced by the regularization: optimal feature / kernel weights are requested to be small. This is easier to achieve for features (or entire feature spaces, as implied by kernels) that are scaled to be of large magnitude, while downscaling them would require a correspondingly upscaled weight for representing the same predictive model. Upscaling (downscaling) features is thus equivalent to modifying regularizers such that they penalize those features less (more). As is common practice, we here use isotropic regularizers, which penalize all dimensions uniformly. This implies that the kernels have to be normalized in a sensible way in order to represent an "uninformative prior" as to which kernels are useful.

There exist several approaches to kernel normalization, of which we use two in the computational experiments below. They are fundamentally different. The first one generalizes the common practice of standardizing features to entire kernels, thereby directly implementing the spirit of the discussion above. In contrast, the second normalization approach rescales the data points to unit norm in feature space. Nevertheless it can have a beneficial effect on the scaling of kernels, as we argue below.

Multiplicative Normalization. As done in Ong and Zien (2008), we multiplicatively normalize the kernels to have uniform variance of data points in feature space. Formally, we find a positive rescaling ρ_m of the kernel, such that the rescaled kernel $\tilde{k}_m(\cdot,\cdot) = \rho_m k_m(\cdot,\cdot)$ and the corresponding feature map $\tilde{\Phi}_m(\cdot) = \sqrt{\rho_m}\Phi_m(\cdot)$ satisfy

$$\frac{1}{n}\sum_{i=1}^{n} \left\| \tilde{\Phi}_m(x_i) - \tilde{\Phi}_m(\overline{x}) \right\|^2 = 1$$

11. *Large scale* in the sense, that the data cannot be stored in memory or the computation reaches a maintainable limit. In the case of MKL this can be due both a large sample size or a high number of kernels.

for each $m = 1, \ldots, M$, where $\tilde{\Phi}_m(\overline{x}) := \frac{1}{n}\sum_{i=1}^n \tilde{\Phi}_m(x_i)$ is the empirical mean of the data in feature space. The above equation can be equivalently be expressed in terms of kernel functions as

$$\frac{1}{n}\sum_{i=1}^n \tilde{k}_m(x_i,x_i) - \frac{1}{n^2}\sum_{i=1}^n\sum_{j=1}^n \tilde{k}_m(x_i,x_j) = 1,$$

so that the final normalization rule is

$$k(x,\overline{x}) \longmapsto \frac{k(x,\overline{x})}{\frac{1}{n}\sum_{i=1}^n k(x_i,x_i) - \frac{1}{n^2}\sum_{i,j=1}^n, k(x_i,x_j)}.$$

Note that in case the kernel is centered (i.e., the empirical mean of the data points lies on the origin), the above rule simplifies to $k(x,\overline{x}) \longmapsto k(x,\overline{x})/\frac{1}{n}\mathrm{tr}(K)$, where $\mathrm{tr}(K) := \sum_{i=1}^n k(x_i,x_i)$ is the trace of the kernel matrix K.

Spherical Normalization. Frequently, kernels are normalized according to

$$k(x,\overline{x}) \longmapsto \frac{k(x,\overline{x})}{\sqrt{k(x,x)k(\overline{x},\overline{x})}}. \tag{23}$$

After this operation, $\|x\| = k(x,x) = 1$ holds for each data point x; this means that each data point is rescaled to lie on the unit sphere. Still, this also may have an effect on the scale of the features: a spherically normalized and centered kernel is also always multiplicatively normalized, because the multiplicative normalization rule becomes $k(x,\overline{x}) \longmapsto k(x,\overline{x})/\frac{1}{n}\mathrm{tr}(K) = k(x,\overline{x})/1$.

Thus the spherical normalization may be seen as an approximate to the above multiplicative normalization and may be used as a substitute for it. Note, however, that it changes the data points themselves by eliminating length information; whether this is desired or not depends on the learning task at hand. Finally note that both normalizations achieve that the optimal value of C is not far from 1.

4.5 Limitations and Extensions of our Framework

In this section, we show the connection of ℓ_p-norm MKL to a formulation based on block norms, point out limitations and sketch extensions of our framework. To this aim let us recall the primal MKL problem (14) and consider the special case of ℓ_p-norm MKL given by

$$\inf_{w,b,\theta:\theta\geq\mathbf{0}} \quad C\sum_{i=1}^n V\left(\sum_{m=1}^M \langle w_m, \psi_m(x_i)\rangle_{\mathcal{H}_m} + b,\, y_i\right) + \frac{1}{2}\sum_{m=1}^M \frac{\|w_m\|^2_{\mathcal{H}_m}}{\theta_m}, \quad \text{s.t.} \quad \|\theta\|_p^2 \leq 1. \tag{24}$$

The subsequent proposition shows that (24) equivalently can be translated into the following mixed-norm formulation,

$$\inf_{w,b} \quad \tilde{C}\sum_{i=1}^n V\left(\sum_{m=1}^M \langle w_m, \psi_m(x_i)\rangle_{\mathcal{H}_m} + b,\, y_i\right) + \frac{1}{2}\sum_{m=1}^M \|w_m\|^q_{\mathcal{H}_m}, \tag{25}$$

where $q = \frac{2p}{p+1}$, and $\tilde{C}$ is a constant. This has been studied by Bach et al. (2004) for $q = 1$ and by Szafranski et al. (2008) for hierarchical penalization.

Proposition 5 *Let be $p > 1$, be V a convex loss function, and define $q := \frac{2p}{p+1}$ (i.e., $p = \frac{q}{2-q}$). Optimization Problem* (24) *and* (25) *are equivalent, that is, for each C there exists a $\tilde{C} > 0$, such that for each optimal solution (w^*, b^*, θ^*) of OP* (24) *using C, we have that (w^*, b^*) is also optimal in OP* (25) *using $\tilde{C}$, and vice versa.*

Proof From Prop. 2 it follows that for any fixed w in (24) it holds for the w-optimal θ:

$$\exists \zeta: \quad \theta_m = \zeta \|w_m\|_{\mathcal{H}_m}^{\frac{2}{p+1}}, \quad \forall m = 1, \ldots, M.$$

Plugging the above equation into (24) yields

$$\inf_{w,b} \quad C \sum_{i=1}^{n} V\left(\sum_{m=1}^{M} \langle w_m, \psi_m(x_i) \rangle_{\mathcal{H}_m} + b,\, y_i \right) + \frac{1}{2\zeta} \sum_{m=1}^{M} \|w_m\|_{\mathcal{H}_m}^{\frac{2p}{p+1}}.$$

Defining $q := \frac{2p}{p+1}$ and $\tilde{C} := \zeta C$ results in (25). ■

Now, let us take a closer look on the parameter range of q. It is easy to see that when we vary p in the real interval $[1, \infty]$, then q is limited to range in $[1, 2]$. So in other words the methodology presented in this paper only covers the $1 \leq q \leq 2$ block norm case. However, from an algorithmic perspective our framework can be easily extended to the $q > 2$ case: although originally aiming at the more sophisticated case of hierarchical kernel learning, Aflalo et al. (2011) showed in particular that for $q \geq 2$, Equation (25) is equivalent to

$$\sup_{\theta: \theta \geq \mathbf{0}, \|\theta\|_r^2 \leq 1} \inf_{w,b} \quad \tilde{C} \sum_{i=1}^{n} V\left(\sum_{m=1}^{M} \langle w_m, \psi_m(x_i) \rangle_{\mathcal{H}_m} + b,\, y_i \right) + \frac{1}{2} \sum_{m=1}^{M} \theta_m \|w_m\|_{\mathcal{H}_m}^2,$$

where $r := \frac{q}{q-2}$. Note the difference to ℓ_p-norm MKL: the mixing coefficients θ appear in the nominator and by varying r in the interval $[1, \infty]$, the range of q in the interval $[2, \infty]$ can be obtained, which explains why this method is complementary to ours, where q ranges in $[1, 2]$.

It is straightforward to show that for every fixed (possibly suboptimal) pair (w, b) the optimal θ is given by

$$\theta_m = \frac{\|w_m\|_{\mathcal{H}_m}^{\frac{2}{r-1}}}{\left(\sum_{m'=1}^{M} \|w_{m'}\|_{\mathcal{H}_{m'}}^{\frac{2r}{r-1}} \right)^{1/r}}, \quad \forall m = 1, \ldots, M.$$

The proof is analogous to that of Prop. 2 and the above analytical update formula can be used to derive a block coordinate descent algorithm that is analogous to ours. In our framework, the mixings θ, however, appear in the denominator of the objective function of Optimization Problem (14). Therefore, the corresponding update formula in our framework is

$$\theta_m = \frac{\|w_m\|_{\mathcal{H}_m}^{\frac{-2}{r-1}}}{\left(\sum_{m'=1}^{M} \|w_{m'}\|_{\mathcal{H}_{m'}}^{\frac{-2r}{r-1}} \right)^{1/r}}, \quad \forall m = 1, \ldots, M. \tag{26}$$

This shows that we can simply optimize $2 < q \leq \infty$-block-norm MKL within our computational framework, using the update formula (26).

5. Theoretical Analysis

In this section we present a theoretical analysis of ℓ_p-norm MKL, based on Rademacher complexities.[12] We prove a theorem that converts any Rademacher-based generalization bound on ℓ_1-norm MKL into a generalization bound for ℓ_p-norm MKL (and even more generally: arbitrary-norm MKL). Remarkably this ℓ_1-to-ℓ_p conversion is obtained almost without any effort: by a simple 5-line proof. The proof idea is based on Kloft et al. (2010). We remark that an ℓ_p-norm MKL bound was already given in Cortes et al. (2010a), but their bound is only valid for the special cases where $p/(p-1)$ is an integer and is not tight for small p, as it diverges to infinity when $p > 1$ and p approaches one. By contrast, beside a negligible $\log(M)$-factor, our result matches the best known lower bounds, when p approaches one.

Let us start by defining the hypothesis set that we want to investigate. Following Cortes et al. (2010a), we consider the following hypothesis class for $p \in [1,\infty]$:

$$H_M^p := \left\{ h : \mathcal{X} \to \mathbb{R} \;\middle|\; h(x) = \sum_{m=1}^{M} \sqrt{\theta_m} \langle w_m, \psi_m(x) \rangle_{\mathcal{H}_m},\ \|w\|_{\mathcal{H}} \le 1,\ \|\theta\|_p \le 1 \right\}.$$

Solving our primal MKL problem (14) corresponds to empirical risk minimization in the above hypothesis class. We are thus interested in bounding the generalization error of the above class w.r.t. an i.i.d. sample $(x_1,y_1),\ldots,(x_n,y_n) \in \mathcal{X} \times \{-1,1\}$ from an arbitrary distribution P. In order to do so, we compute the *Rademacher complexity*,

$$\mathscr{R}(H_M^p) := \mathbb{E}\left[\sup_{h \in H_M^p} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(x_i) \right],$$

where $\sigma_1,\ldots,\sigma_n$ are independent Rademacher variables (i.e., they obtain the values -1 or +1 with the same probability 0.5) and the $\mathbb{E}$ is the expectation operator that removes the dependency on all random variables, that is, σ_i, x_i, and y_i ($i = 1,\ldots,n$). If the Rademacher complexity is known, there is a large body of results that can be used to bound the generalization error (e.g., Koltchinskii and Panchenko, 2002; Bartlett and Mendelson, 2002).

We now show a simple ℓ_q-to-ℓ_p conversion technique for the Rademacher complexity, which is the main result of this section:

Theorem 6 (ℓ_q-to-ℓ_p Conversion) *For any sample of size n and $1 \le q \le p \le \infty$ the Rademacher complexity of the hypothesis set H_M^p can be bounded in terms of H_M^q,*

$$\mathscr{R}(H_M^p) \le \sqrt{M^{\frac{1}{q}-\frac{1}{p}}}\, \mathscr{R}(H_M^q).$$

In particular, we have $\mathscr{R}(H_M^p) \le \sqrt{M^{1/p^*}}\, \mathscr{R}(H_M^1)$ (ℓ_1-to-ℓ_p Conversion), where $p^* := p/(p-1)$ is the conjugated exponent of p.

Proof By Hölder's inequality (e.g., Steele, 2004), denoting $\theta^p := (\theta_1^p,\cdots,\theta_M^p)^\top$, we have for all non-negative $\theta \in \mathbb{R}^M$,

$$\|\theta\|_q = (\mathbf{1}^\top \theta^q)^{1/q} \le \left(\|\mathbf{1}\|_{(p/q)^*} \|\theta^q\|_{p/q} \right)^{1/q} = M^{\frac{1}{q(p/q)^*}} \|\theta\|_p = M^{\frac{1}{q}-\frac{1}{p}} \|\theta\|_p \,. \tag{27}$$

12. An introduction to statistical learning theory, which may equip the reader with the needed notions used in this section, is given in Bousquet et al. (2004). See also, for example, Section 4 in Shawe-Taylor and Cristianini (2004).

Hence,

$$\begin{aligned}
\mathscr{R}(H_M^p) &\overset{\text{Def.}}{=} \mathbb{E}\left[\sup_{w:\|w\|_{\mathcal{H}}\le 1,\ \theta:\|\theta\|_p\le 1} \frac{1}{n}\sum_{i=1}^n \sigma_i \sum_{m=1}^M \sqrt{\theta_m}\langle w_m, \psi_m(x_i)\rangle_{\mathcal{H}_m}\right] \\
&\overset{(27)}{\le} \mathbb{E}\left[\sup_{w:\|w\|_{\mathcal{H}}\le 1,\ \theta:\|\theta\|_q\le M^{\frac{1}{q}-\frac{1}{p}}} \frac{1}{n}\sum_{i=1}^n \sigma_i \sum_{m=1}^M \sqrt{\theta_m}\langle w_m, \psi_m(x_i)\rangle_{\mathcal{H}_m}\right] \\
&= \mathbb{E}\left[\sup_{w:\|w\|_{\mathcal{H}}\le 1,\ \theta:\|\theta\|_q\le 1} \frac{1}{n}\sum_{i=1}^n \sigma_i \sum_{m=1}^M \sqrt{\theta_m M^{\frac{1}{q}-\frac{1}{p}}}\langle w_m, \psi_m(x)\rangle_{\mathcal{H}_m}\right] \\
&\overset{\text{Def.}}{=} \sqrt{M^{\frac{1}{q}-\frac{1}{p}}}\,\mathscr{R}(H_M^q).
\end{aligned}$$

■

Remark 7 *More generally we have that for any norm $\|\cdot\|_\star$ on $\mathbb{R}^M$, because all norms on $\mathbb{R}^M$ are equivalent (e.g., Rudin, 1991), there exists a $c_\star \in \mathbb{R}$ such that*

$$\mathscr{R}(H_M^p) \le c_\star \mathscr{R}(H_M^\star).$$

This means the conversion technique extends to arbitrary norms: for any given norm $\|\cdot\|_\star$, we can convert any bound on $\mathscr{R}(H_M^p)$ into a bound on the Rademacher complexity $\mathscr{R}(H_M^\star)$ of hypothesis set induced by $\|\cdot\|_\star$.

A nice characteristic of the above result is that we can make use of any existing bound on the Rademacher complexity of H_M^1 in order to obtain a generalization bound for H_M^p. This fact is illustrated in the following. For example, it has recently been shown:

Theorem 8 (Cortes et al., 2010a) *Let $M > 1$ and assume that $k_m(x,x) \le R^2$ for all $x \in \mathcal{X}$ and $m = 1,\ldots,M$. Then, for any sample of size n, the Rademacher complexities of the hypothesis sets H_M^1 and H_M^p can be bounded as follows (where $c := 23/22$ and $\lceil\cdot\rceil$ rounds to the next largest integer):*

$$\mathscr{R}(H_M^1) \le \sqrt{\frac{ce\lceil\log M\rceil R^2}{n}}, \quad \mathscr{R}(H_M^p) \le \sqrt{\frac{cp^* M^{1/p^*} R^2}{n}},$$

for any $p > 1$ such that p^* is an even integer.

For $p = 1$ [$p > 1$] the above result directly leads to a $O(\sqrt{\log M})$ $\left[O(\sqrt{M^{1/p^*}})\right]$ bound on the generalization error and thus substantially improves on a series of loose results given within the past years (see Cortes et al., 2010a, and references therein). Unfortunately, since p^* is required to be an integer, the range of p is restricted to $p \in [1,2]$. As a remedy, in this paper we use the ℓ_q-to-ℓ_p Conversion technique to the above result[13] to obtain a bound for H_M^p that holds for all $p \in [1,\ldots,\infty]$: the following corollary is obtained from the previous theorem by using ℓ_q-to-ℓ_p-norm conversion for $q = 1$ and $q = \lceil p^*\rceil^*$, respectively, and then taking the minimum value of the so-obtained bounds.

13. The point here is that we could use any ℓ_1-bound, for example, the bounds of Kakade et al. (2009) and Kloft et al. (2010) have the same favorable $O(\log M)$ rate; in particular, whenever a new ℓ_1-bound is proven, we can plug it into our conversion technique to obtain a new bound.

Corollary 9 (of the previous two theorems) *Let $M > 1$ and assume that $k_m(x,x) \le R^2$ for all $x \in \mathcal{X}$ and $m = 1,\ldots,M$. Then, for any sample of size n, the Rademacher complexity of the hypothesis set H_M^p can be bounded as follows:*

$$\forall p \in [1,...,\infty]: \quad \mathscr{R}(H_M^p) \le \sqrt{\frac{cM^{1/p^*}R^2 \min\left(e\lceil \log M\rceil, \lceil p^*\rceil\right)}{n}},$$

where $p^* := p/(p-1)$ is the conjugated exponent of p and $c := 23/22$.

It is instructive to compare the above bound, which we obtained by our ℓ_q-to-ℓ_p conversion technique, with the one given in Cortes et al. (2010a): that is $\mathscr{R}(H_M^p) \le \sqrt{\frac{cep^*M^{1/p^*}R^2}{n}}$ for any $p \in [1,...,\infty]$ such that p^* is an integer. First, we observe that for $p = 2$ the bounds' rates coincide. Second, we observe that for small p (close to one), the p^*-factor in the Cortes-bound leads to considerably high constants. When p approaches one, it even diverges to infinity. In contrast, our bound converges to $\mathscr{R}(H_M^p) \le \sqrt{\frac{ce\lceil \log M\rceil R^2}{n}}$ when p approaches one, which is precisely the tight 1-norm bound of Thm. 8. Finally, it is also interesting to consider the case $p \ge 2$ (which is not covered by the Cortes et al., 2010a bound): if we let $p \to \infty$, we obtain $\mathscr{R}(H_M^p) \le \sqrt{\frac{2cMR^2}{n}}$. This matches the well-known $O\left(\sqrt{M}\right)$ lower bounds based on the VC-dimension (e.g., Devroye et al., 1996, Section 14).

We now make use of the above analysis of the Rademacher complexity to bound the generalization error. There are many results in the literature that can be employed to this aim. Ours is based on Thm. 7 in Bartlett and Mendelson (2002):

Corollary 10 *Let $M > 1$ and $p \in]1,...,\infty]$. Assume that $k_m(x,x) \le R^2$ for all $x \in \mathcal{X}$ and $m = 1,\ldots,M$. Assume the loss $V : \mathbb{R} \to [0,1]$ is Lipschitz with constant L and $V(t) \ge 1$ for all $t \le 0$. Set $p^* := p/(p-1)$ and $c := 23/22$. Then, the following holds with probability larger than $1-\delta$ over samples of size n for all classifiers $h \in H_M^p$:*

$$R(h) \le \widehat{R}(h) + 4L\sqrt{\frac{cM^{1/p^*}R^2 \min\left(e\lceil \log M\rceil, \lceil p^*\rceil\right)}{n}} + \sqrt{\frac{\ln(2/\delta)}{2n}},$$

where $R(h) = \mathrm{P}\left[yh(x) \le 0\right]$ is the expected risk w.r.t. 0-1 loss and $\widehat{R}(h) = \frac{1}{n}\sum_{i=1}^n V(y_i h(x_i))$ is the empirical risk w.r.t. loss V.

The above theorem is formulated for general Lipschitz loss functions. Since the margin loss $V(t) = \min\left(1, [1-t/\gamma]_+\right)$ is Lipschitz with constant $1/\gamma$ and upper bounding the 0-1 loss, it fulfills the preliminaries of the above corollary. Hence, we immediately obtain the following radius-margin bound (see also Koltchinskii and Panchenko, 2002):

Corollary 11 (ℓ_p-norm MKL Radius-Margin Bound) *Fix the margin $\gamma > 0$. Let $M > 1$ and $p \in]1,...,\infty]$. Assume that $k_m(x,x) \le R^2$ for all $x \in \mathcal{X}$ and $m = 1,\ldots,M$. Set $p^* := p/(p-1)$ and $c := 23/22$. Then, the following holds with probability larger than $1-\delta$ over samples of size n for all classifiers $h \in H_M^p$:*

$$R(h) \le \widehat{R}(h) + \frac{4R}{\gamma}\sqrt{\frac{cM^{1/p^*} \min\left(e\lceil \log M\rceil, \lceil p^*\rceil\right)}{n}} + \sqrt{\frac{\ln(2/\delta)}{2n}},$$

where $R(h) = \mathrm{P}\big[yh(x) \leq 0\big]$ *is the expected risk w.r.t. 0-1 loss and* $\widehat{R}(h) = \frac{1}{n}\sum_{i=1}^{n} \min\big(1, [1 - y_i h(x_i)/\gamma]_+\big)$ *the empirical risk w.r.t. margin loss.*

Finally, we would like to point out that, for reasons stated in Remark 7, the ℓ_q-to-ℓ_p conversion technique can be extended to norms different than ℓ_p . This lets us extend the above bounds to, for example, block norms and sums of block norms as used in elastic-net regularization (see Kloft et al., 2010, for such bounds), but also non-isotropic norms such as weighted ℓ_p-norms.

5.1 Case-based Analysis of a Sparse and a Non-Sparse Scenario

From the results given in the last section it seems that it is beneficial to use a sparsity-inducing ℓ_1-norm penalty when learning with multiple kernels. This however somewhat contradicts our empirical evaluation, which indicated that the optimal norm parameter p depends on the true underlying sparsity of the problem. Indeed, as we show below, a refined theoretical analysis supports this intuitive claim. We show for an exemplary scenario that if the underlying truth is uniformly non-sparse, then a non-sparse ℓ_p-norm is more promising than a sparse one. On the other hand, we illustrate that in a sparse scenario, the sparsity-inducing ℓ_1-norm indeed can be beneficial.

We start by reparametrizing our hypothesis set based on block norms: by Prop. 5 it holds that

$$H_M^p = \left\{ h : \mathcal{X} \to \mathbb{R} \;\middle|\; h(x) = \sum_{m=1}^{M} \langle w_m, \psi_m(x) \rangle_{\mathcal{H}_m},\ \|w\|_{2,q} \leq 1,\ q := 2p/(p+1) \right\},$$

where $||w||_{2,q} := \left(\sum_{m=1}^{M} ||w_m||_{\mathcal{H}_m}^q\right)^{1/q}$ is the $\ell_{2,q}$-block norm. This means we can equivalently parametrize our hypothesis set in terms of block norms. Second, let us generalize the set by introducing an additional parameter C as follows

$${}^{C}H_M^p := \left\{ h : \mathcal{X} \to \mathbb{R} \;\middle|\; h(x) = \sum_{m=1}^{M} \langle w_m, \psi_m(x) \rangle_{\mathcal{H}_m},\ \|w\|_{2,q} \leq C,\ q := 2p/(p+1) \right\}.$$

Clearly, ${}^{C}H_M^p = H_M^p$ for $C = 1$, which explains why the parametrization via C is more general. It is straightforward to verify that $\mathscr{R}\left({}^{C}H_M^p\right) = C\mathscr{R}\left(H_M^p\right)$ for any C. Hence, under the preliminaries of Corollary 10, we have

$$\begin{aligned} p > 1: \qquad & R(h) \leq \widehat{R}(h) + 4L\sqrt{\frac{cM^{1/p^*}R^2C^2\min\left(e\lceil \log M \rceil, \lceil p^* \rceil\right)}{n}} + \sqrt{\frac{\ln(2/\delta)}{2n}}, \\ p = 1: \qquad & R(h) \leq \widehat{R}(h) + 4L\sqrt{\frac{ce\lceil \log M \rceil R^2C^2}{n}} + \sqrt{\frac{\ln(2/\delta)}{2n}}\ . \end{aligned} \tag{28}$$

We will exploit the above bound in the following two illustrate examples.

Example 1. Let the input space be $\mathcal{X} = \mathbb{R}^M$, and the feature map be $\psi_m(x) = x_m$ for all $m = 1, \ldots, M$ and $x = (x_1, ..., x_M) \in \mathcal{X}$ (in other words, ψ_m is a projection on the mth feature). Assume that the Bayes-optimal classifier is given by

$$w_{\text{Bayes}} = (1, \ldots, 1)^\top \in \mathbb{R}^M.$$

This means the best classifier possible is uniformly non-sparse (see Fig. 1, left). Clearly, it can be advantageous to work with a hypothesis set that is rich enough to contain the Bayes classifier,

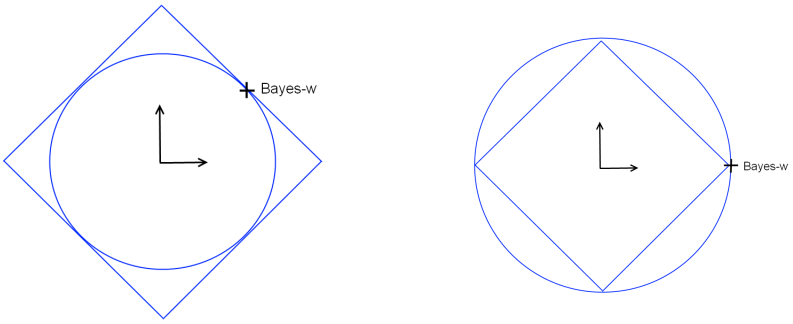

Figure 1: Illustration of the two analyzed cases for $M = 2$: a uniformly non-sparse (Example 1, left) and a sparse (Example 2, right) Scenario.

that is, $(1,\ldots,1)^\top \in {}^C H_M^p$. In our example, this is the case if and only if $\|(1,\ldots,1)^\top\|_{2p/(p+1)} \leq C$, which itself is equivalent to $M^{(p+1)/2p} \leq C$. The bound (28) attains its minimal value under the latter constraint for $M^{(p+1)/2p} = C$. Resubstitution into the bound yields

$$p > 1: \qquad R(h) \leq \widehat{R}(h) + 4L\sqrt{\frac{cM^2R^2\min\left(e\lceil\log M\rceil, \lceil p^*\rceil\right)}{n}} + \sqrt{\frac{\ln(2/\delta)}{2n}}.$$

$$p = 1: \qquad R(h) \leq \widehat{R}(h) + 4L\sqrt{\frac{ceM^2\lceil\log M\rceil R^2C^2}{n}} + \sqrt{\frac{\ln(2/\delta)}{2n}}\,.$$

Let us now compare the so obtained rate: for $p > 1$ we get $O(M^2)$ and for $p = 1$ we have $O(M^2\log(M))$. So the rates differ by a $\log(M)$ factor. This means that in this particular (non-sparse) example, neglecting the constants, the non-sparse $p > 1$-norm MKL variants yield a strictly better generalization bound than ℓ_1-norm MKL.

Example 2. In this second example we consider the same input space and kernels as before. But this time we assume a *sparse* Bayes-optimal classifier (see Fig. 1, right)

$$w_{\text{Bayes}} = (1,0,\ldots,0)^\top \in \mathbb{R}^M.$$

As in the previous example, in order w_{Bayes} to be in the hypothesis set, we have to require $\|(1,0,\ldots,0)^\top\|_{2p/(p+1)} \leq C$. But this time this simply solves to $C \geq 1$, which is independent of the norm parameter p. Thus, inserting $C = 1$ in the bound (28), we obtain

$$p > 1: \qquad R(h) \leq \widehat{R}(h) + 4L\sqrt{\frac{cM^2R^2\min\left(e\lceil\log M\rceil, \lceil p^*\rceil\right)}{n}} + \sqrt{\frac{\ln(2/\delta)}{2n}}.$$

$$p = 1: \qquad R(h) \leq \widehat{R}(h) + 4L\sqrt{\frac{ceM^2\lceil\log M\rceil R^2}{n}} + \sqrt{\frac{\ln(2/\delta)}{2n}}\,.$$

Clearly, in this particular sparse example, the $\ell_{p=1}$-bound is considerably smaller than the one of $\ell_{p>1}$-norm MKL—especially, if the number of kernels is high compared to the sample size. This is also intuitive: if the underlying truth is sparse, we expect a sparsity-inducing norm to match well the ground truth.

We conclude from the previous two examples that the optimal norm parameter p depends on the underlying ground truth: if it is sparse, then choosing a sparse regularization is beneficial; otherwise, a non-sparse norm p can perform well. This is somewhat contrary to anecdotal reports, which claim that sparsity-inducing norms are beneficial in high (kernel) dimensions. This is because those analyses implicitly assume the ground truth to be sparse. The present paper, however, clearly shows that we might encounter a non-sparse ground truth in practical applications (see experimental section).

6. Computational Experiments

In this section we study non-sparse MKL in terms of computational efficiency and predictive accuracy. We apply the method of Sonnenburg et al. (2006a) in the case of $p = 1$. We write ℓ_∞-norm MKL for a regular SVM with the unweighted-sum kernel $K = \sum_m K_m$.

We first study a toy problem in Section 6.1 where we have full control over the distribution of the relevant information in order to shed light on the appropriateness of sparse, non-sparse, and ℓ_∞-MKL. We report on real-world problems from bioinformatics, namely protein subcellular localization (Section 6.2), finding transcription start sites of RNA Polymerase II binding genes in genomic DNA sequences (Section 6.3), and reconstructing metabolic gene networks (Section 6.4). All data sets used in this section were made available online (see supplementary homepage of this paper: `http://doc.ml.tu-berlin.de/nonsparse_mkl/`).

6.1 Measuring the Impact of Data Sparsity—Toy Experiment

The goal of this section is to study the relationship of the level of sparsity of the true underlying function to be learned to the chosen norm p in the model. Intuitively, we might expect that the optimal choice of p directly corresponds to the true level of sparsity. Apart from verifying this conjecture, we are also interested in the effects of suboptimal choice of p. To this aim we constructed several artificial data sets in which we vary the degree of sparsity in the true kernel mixture coefficients. We go from having all weight focused on a single kernel (the highest level of sparsity) to uniform weights (the least sparse scenario possible) in several steps. We then study the statistical performance of ℓ_p-norm MKL for different values of p that cover the entire range $[1,\infty]$.

We generated a data set as follows (we made this so-called *mkl-toy* data set available at the mldata repository[14]). An n-element balanced sample $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ is generated from two $d = 50$-dimensional isotropic Gaussian distributions with equal covariance matrices $C = I_{d\times d}$ and equal, but opposite, means $\mu_1 = \frac{\rho}{\|\theta\|_2}\theta$ and $\mu_2 = -\mu_1$. Thereby θ is a binary vector, that is, $\forall i : \theta_i \in \{0,1\}$, encoding the true underlying data sparsity as follows. Zero components $\theta_i = 0$ clearly imply identical means of the two classes' distributions in the ith feature set; hence the latter does not carry any discriminating information. In summary, the fraction of zero components, $\nu(\theta) = 1 - \frac{1}{d}\sum_{i=1}^d \theta_i$, is a measure for the feature sparsity of the learning problem.

For $\nu \in \{0, 0.44, 0.64, 0.82, 0.92, 1\}$ we generate six data sets $\mathcal{D}_1, \ldots, \mathcal{D}_6$ fixing $\rho = 1.75$. Then, each feature is input to a linear kernel and the resulting kernel matrices are multiplicatively normalized as described in Section 4.4.2. Hence, $\nu(\theta)$ gives the fraction of noise kernels in the working kernel set. Then, classification models are computed by training ℓ_p-norm MKL for $p = 1, 4/3, 2, 4, \infty$ on each $\mathcal{D}_i$. Soft margin parameters C are tuned on independent $10,000$-elemental validation sets

14. The repository can be found at `http://mldata.org/repository/data/viewslug/mkl-toy/`.

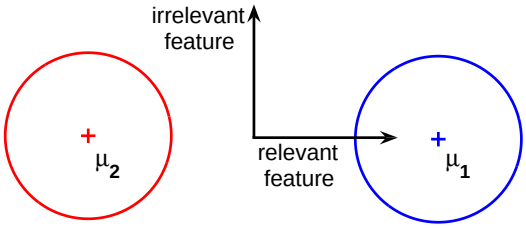

Figure 2: Illustration of the toy experiment for $\theta = (1,0)^\top$.

by grid search over $C \in 10^{[-4,3.5,\ldots,0]}$ (optimal Cs are attained in the interior of the grid). The relative duality gaps were optimized up to a precision of 10^{-3}. We report on test errors evaluated on $10,000$-elemental independent test sets and pure mean ℓ_2 model errors of the computed kernel mixtures, that is $\mathrm{ME}(\widehat{\theta}) = \|\zeta(\widehat{\theta}) - \zeta(\theta)\|_2$, where $\zeta(x) = \frac{x}{\|x\|_2}$.

The results are shown in Fig. 3 for $n = 50$ and $n = 800$, where the figures on the left show the test errors and the ones on the right the model errors $\mathrm{ME}(\widehat{\theta})$. Regarding the latter, model errors reflect the corresponding test errors for $n = 50$. This observation can be explained by statistical learning theory. The minimizer of the empirical risk performs unstable for small sample sizes and the model selection results in a strongly regularized hypothesis, leading to the observed agreement between test error and model error.

Unsurprisingly, ℓ_1 performs best and reaches the Bayes error in the sparse scenario, where only a single kernel carries the whole discriminative information of the learning problem. However, in the other scenarios it mostly performs worse than the other MKL variants. This is remarkable because the underlying ground truth, that is, the vector θ, is sparse in all but the uniform scenario. In other words, selecting this data set may imply a bias towards ℓ_1-norm. In contrast, the vanilla SVM using an unweighted sum kernel performs best when all kernels are equally informative, however, its performance does not approach the Bayes error rate. This is because it corresponds to a $\ell_{2,2}$-block norm regularization (see Sect. 4.5) but for a truly uniform regularization a ℓ_∞-block norm penalty (as employed in Nath et al., 2009) would be needed. This indicates a limitation of our framework; it shall, however, be kept in mind that such a uniform scenario might quite artificial. The non-sparse ℓ_4- and ℓ_2-norm MKL variants perform best in the balanced scenarios, that is, when the noise level is ranging in the interval 64%-92%. Intuitively, the non-sparse ℓ_4-norm MKL is the most robust MKL variant, achieving a test error of less than 10% in all scenarios. Tuning the sparsity parameter p for each experiment, ℓ_p-norm MKL achieves the lowest test error across all scenarios.

When the sample size is increased to $n = 800$ training instances, test errors decrease significantly. Nevertheless, we still observe differences of up to 1% test error between the best (ℓ_∞-norm MKL) and worst (ℓ_1-norm MKL) prediction model in the two most non-sparse scenarios. Note that all ℓ_p-norm MKL variants perform well in the sparse scenarios. In contrast with the test errors, the mean model errors depicted in Figure 3 (bottom, right) are relatively high. Similarly to above reasoning, this discrepancy can be explained by the minimizer of the empirical risk becoming stable when increasing the sample size, which decreases the generalization error (see theoretical Analysis in Section 5, where it was shown that the speed of the minimizer becoming stable is at least of a rate of $O(1/\sqrt{n})$). Again, ℓ_p-norm MKL achieves the smallest test error for all scenarios for appropriately chosen p and for a fixed p across all experiments, the non-sparse ℓ_4-norm MKL performs the most robustly.

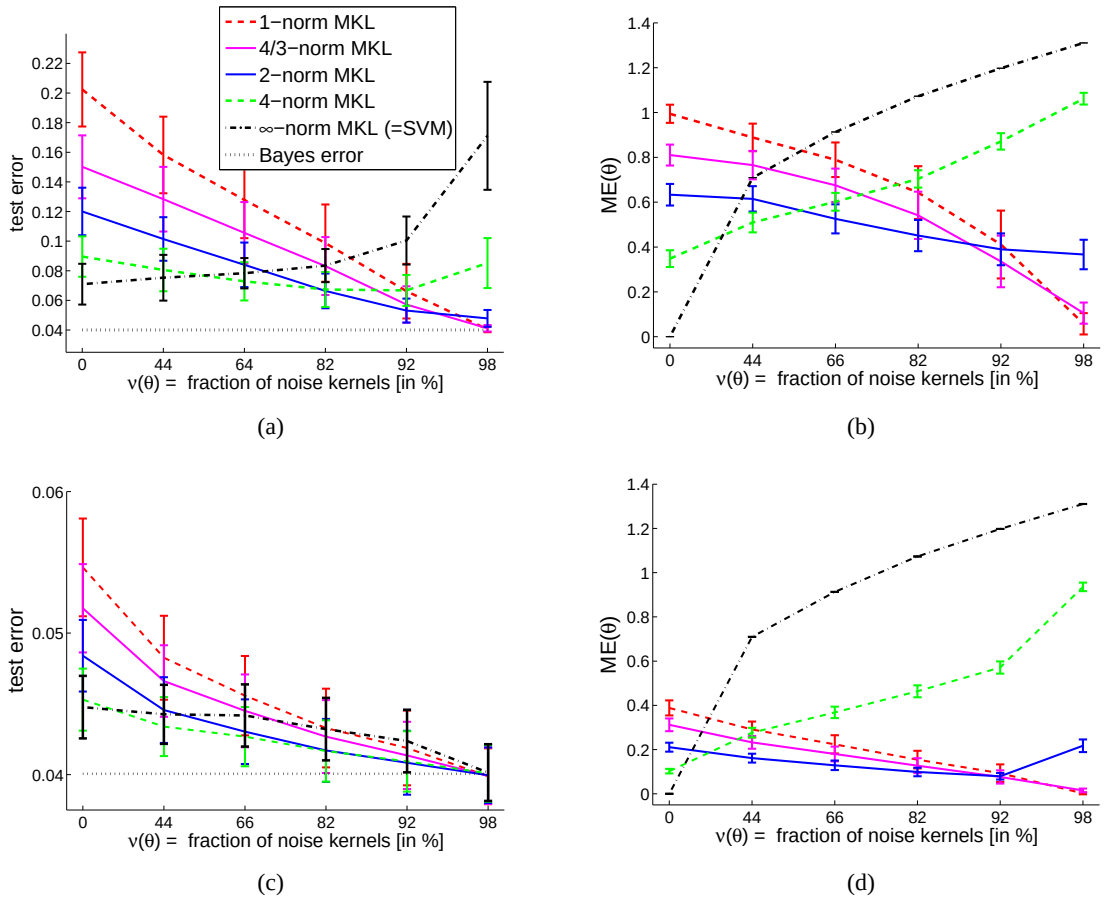

Figure 3: Results of the artificial experiment for sample sizes of $n = 50$ (top) and $n = 800$ (below) training instances in terms of test errors (left) and mean ℓ_2 model errors $\mathrm{ME}(\widehat{\theta})$ (right).

In summary, the choice of the norm parameter p is important for small sample sizes, whereas its impact decreases with an increase of the training data. As expected, sparse MKL performs best in sparse scenarios, while non-sparse MKL performs best in moderate or non-sparse scenarios, and for uniform scenarios the unweighted-sum kernel SVM performs best. For appropriately tuning the norm parameter, ℓ_p-norm MKL proves robust in all scenarios.

6.2 Protein Subcellular Localization—A Sparse Scenario

The prediction of the subcellular localization of proteins is one of the rare empirical success stories of ℓ_1-norm-regularized MKL (Ong and Zien, 2008; Zien and Ong, 2007): after defining 69 kernels that capture diverse aspects of protein sequences, ℓ_1-norm-MKL could raise the predictive accuracy significantly above that of the unweighted sum of kernels, and thereby also improve on established prediction systems for this problem. This has been demonstrated on 4 data sets, corresponding to 4 different sets of organisms (plants, non-plant eukaryotes, Gram-positive and Gram-negative

ℓ_p-**norm**	1	32/31	16/15	8/7	4/3	2	4	8	16	∞
plant	**8.18**	8.22	8.20	8.21	8.43	9.47	11.00	11.61	11.91	11.85
std. err.	$\pm$0.47	$\pm$0.45	$\pm$0.43	$\pm$0.42	$\pm$0.42	$\pm$0.43	$\pm$0.47	$\pm$0.49	$\pm$0.55	$\pm$0.60
nonpl	**8.97**	9.01	9.08	9.19	9.24	9.43	9.77	10.05	10.23	10.33
std. err.	$\pm$0.26	$\pm$0.25	$\pm$0.26	$\pm$0.27	$\pm$0.29	$\pm$0.32	$\pm$0.32	$\pm$0.32	$\pm$0.32	$\pm$0.31
psortNeg	9.99	9.91	**9.87**	10.01	10.13	11.01	12.20	12.73	13.04	13.33
std. err.	$\pm$0.35	$\pm$0.34	$\pm$0.34	$\pm$0.34	$\pm$0.33	$\pm$0.32	$\pm$0.32	$\pm$0.34	$\pm$0.33	$\pm$0.35
psortPos	13.07	**13.01**	13.41	13.17	13.25	14.68	15.55	16.43	17.36	17.63
std. err.	$\pm$0.66	$\pm$0.63	$\pm$0.67	$\pm$0.62	$\pm$0.61	$\pm$0.67	$\pm$0.72	$\pm$0.81	$\pm$0.83	$\pm$0.80

Table 1: Results for Protein Subcellular Localization. For each of the 4 data sets (rows) and each considered norm (columns), we present a measure of prediction error together with its standard error. As measure of prediction error we use 1 minus the average MCC, displayed as percentage.

bacteria) with differing sets of relevant localizations. In this section, we investigate the performance of non-sparse MKL on the same 4 data sets.

The experimental setup used here is related to that of Ong and Zien (2008), although it deviates from it in several details. The kernel matrices are multiplicatively normalized as described in Section 4.4.2. For each data set, we perform the following steps for each of the 30 predefined splits in training set and test set (downloaded from the same URL): We consider norms $p \in \{1, 32/31, 16/15, 8/7, 4/3, 2, 4, 8, \infty\}$ and regularization constants $C \in \{1/32, 1/8, 1/2, 1, 2, 4, 8, 32, 128\}$. For each parameter setting (p, C), we train ℓ_p-norm MKL using a 1-vs-rest strategy on the training set. The predictions on the test set are then evaluated w.r.t. average (over the classes) MCC (Matthews correlation coefficient). As we are only interested in the influence of the norm on the performance, we forbear proper cross-validation (the so-obtained systematical error affects all norms equally). Instead, for each of the 30 data splits and for each p, the value of C that yields the highest MCC is selected. Thus we obtain an optimized C and *MCC* value for each combination of data set, split, and norm p. For each norm, the final *MCC* value is obtained by averaging over the data sets and splits (i.e., C is selected to be optimal for each data set and split).

The results, shown in Table 1, indicate that indeed, with proper choice of a non-sparse regularizer, the accuracy of ℓ_1-norm can be recovered. On the other hand, non-sparse MKL can approximate the ℓ_1-norm arbitrarily close, and thereby approach the same results. However, even when 1-norm is clearly superior to ∞-norm, as for these 4 data sets, it is possible that intermediate norms perform even better. As the table shows, this is indeed the case for the PSORT data sets, albeit only slightly and not significantly so.

We briefly mention that the superior performance of $\ell_{p\approx 1}$-norm MKL in this setup is not surprising. There are four sets of 16 kernels each, in which each kernel picks up very similar information: they only differ in number and placing of gaps in all substrings of length 5 of a given part of the protein sequence. The situation is roughly analogous to considering (inhomogeneous) polynomial kernels of different degrees on the same data vectors. This means that they carry large parts of overlapping information. By construction, also some kernels (those with less gaps) in principle have access to more information (similar to higher degree polynomials including low degree polynomials). Further, Ong and Zien (2008) studied single kernel SVMs for each kernel individually and found that in most cases the 16 kernels from the same subset perform very similarly. This means

that each set of 16 kernels is highly redundant and the excluded parts of information are not very discriminative. This renders a non-sparse kernel mixture ineffective. We conclude that ℓ_1-norm must be the best prediction model.

6.3 Gene Start Recognition—A Weighted Non-Sparse Scenario

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.

Transcription start site finders exploit the fact that the features of promoter regions and the transcription start sites are different from the features of other genomic DNA (Bajic et al., 2004). Many such detectors thereby rely on a combination of feature sets which makes the learning task appealing for MKL. For our experiments we use the data set from Sonnenburg et al. (2006b) which contains a curated set of 8,508 TSS annotated genes using dbTSS version 4 (Suzuki et al., 2002) and refseq genes. These are translated into positive training instances by extracting windows of size $[-1000,+1000]$ around the TSS. Similar to Bajic et al. (2004), 85,042 negative instances are generated from the interior of the gene using the same window size. Following Sonnenburg et al. (2006b), 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 Sonnenburg et al. (2006b). The kernel matrices are spherically normalized as described in section 4.4.2. We reserve 13,000 and 20,000 randomly drawn instances for validation and test sets, respectively, and use the remaining 60,000 as the training pool. Soft margin parameters C are tuned on the validation set by grid search over $C \in 2^{[-2,-1,\ldots,5]}$ (optimal Cs are attained in the interior of the grid). Figure 4 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-norm MKL is significantly outperformed by the sum-kernel. On the contrary, non-sparse MKL significantly achieves higher AUC values than the ℓ_∞-norm 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 (Sonnenburg et al., 2006b) has recently been confirmed to be leading in a comparison of 19 state-of-the-art promoter prediction programs (Abeel et al., 2009), and our experiments suggest that its accuracy can be further elevated by non-sparse MKL.

We give a brief explanation of the reason for optimality of a non-sparse ℓ_p-norm in the above experiments. It has been shown by Sonnenburg et al. (2006b) that there are three highly and two moderately informative kernels. We briefly recall those results by reporting on the AUC performances obtained from training a single-kernel SVM on each kernel individually: TSS signal 0.89, promoter 0.86, 1st exon 0.84, angles 0.55, and energies 0.74, for fixed sample size $n = 2000$. While non-sparse MKL distributes the weights over all kernels (see Fig. 4), sparse MKL focuses on the best kernel. However, the superior performance of non-sparse MKL means that dropping the remaining kernels is detrimental, indicating that they may carry additional discriminative information.

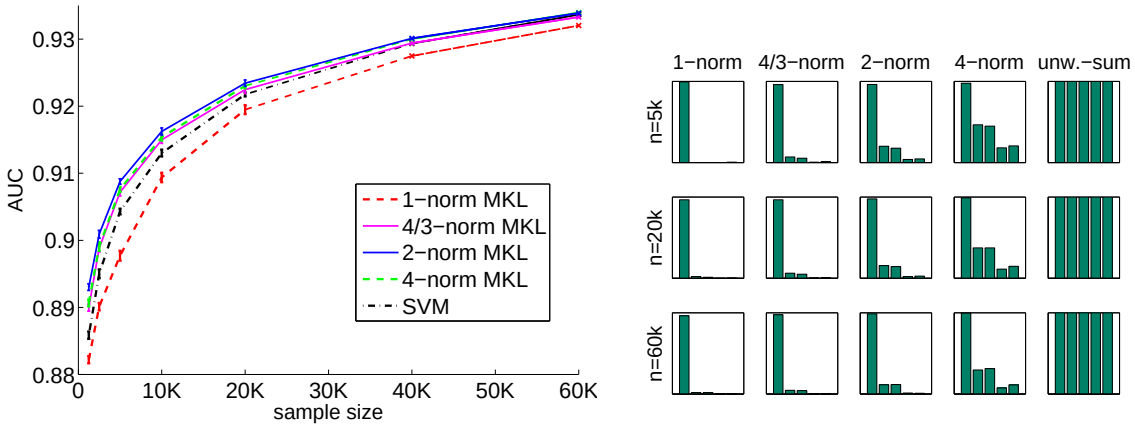

Figure 4: (left) Area under ROC curve (AUC) on test data for TSS recognition as a function of the training set size. Notice the tiny bars indicating standard errors w.r.t. repetitions on disjoint training sets. (right) Corresponding kernel mixtures. For $p = 1$ consistent sparse solutions are obtained while the optimal $p = 2$ distributes weights on the weighted degree and the 2 spectrum kernels in good agreement to Sonnenburg et al. (2006b).

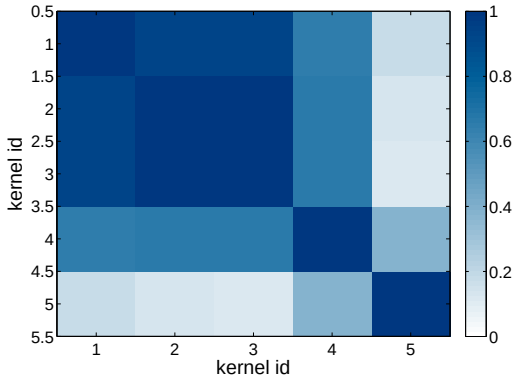

Figure 5: Pairwise alignments of the kernel matrices are shown for the gene start recognition experiment. From left to right, the ordering of the kernel matrices is TSS signal, promoter, 1st exon, angles, and energies. The first three kernels are highly correlated, as expected by their high AUC performances (AUC=0.84–0.89) and the angle kernel correlates decently (AUC=0.55). Surprisingly, the energy kernel correlates only few, despite a descent AUC of 0.74.

To investigate this hypothesis we computed the pairwise alignments of the kernel matrices, that is, $\mathcal{A}(i,j) = \frac{<K_i,K_j>_F}{\|K_i\|_F \|K_j\|_F}$, with respect to the Frobenius dot product (e.g., Golub and van Loan, 1996). The computed alignments are shown in Fig. 5. One can observe that the three relevant kernels are highly aligned as expected since they are correlated via the labels.

However, the energy kernel shows only a slight correlation with the remaining kernels, which is surprisingly little compared to its single kernel performance (AUC=0.74). We conclude that this kernel carries complementary and orthogonal information about the learning problem and should thus be included in the resulting kernel mixture. This is precisely what is done by non-sparse MKL, as can be seen in Fig. 4(right), and the reason for the empirical success of non-sparse MKL on this data set.

6.4 Reconstruction of Metabolic Gene Network—A Uniformly Non-Sparse Scenario

In this section, we apply non-sparse MKL to a problem originally studied by Yamanishi et al. (2005). Given 668 enzymes of the yeast *Saccharomyces cerevisiae* and 2782 functional relationships extracted from the KEGG database (Kanehisa et al., 2004), the task is to predict functional relationships for unknown enzymes. We employ the experimental setup of Bleakley et al. (2007) who phrase the task as graph-based edge prediction with local models by learning a model for each of the 668 enzymes. They provided kernel matrices capturing expression data (EXP), cellular localization (LOC), and the phylogenetic profile (PHY); additionally we use the integration of the former 3 kernels (INT) which matches our definition of an unweighted-sum kernel.

Following Bleakley et al. (2007), we employ a 5-fold cross validation; in each fold we train on average 534 enzyme-based models; however, in contrast to Bleakley et al. (2007) we omit enzymes reacting with only one or two others to guarantee well-defined problem settings. As Table 2 shows, this results in slightly better AUC values for single kernel SVMs where the results by Bleakley et al. (2007) are shown in brackets.

As already observed (Bleakley et al., 2007), the unweighted-sum kernel SVM performs best. Although its solution is well approximated by non-sparse MKL using large values of p, ℓ_p-norm MKL is not able to improve on this $p = \infty$ result. Increasing the number of kernels by including recombined and product kernels does improve the results obtained by MKL for small values of p, but the maximal AUC values are not statistically significantly different from those of ℓ_∞-norm MKL. We conjecture that the performance of the unweighted-sum kernel SVM can be explained by all three kernels performing well individually. Their correlation is only moderate, as shown in Fig. 6, suggesting that they contain complementary information. Hence, downweighting one of those three orthogonal kernels leads to a decrease in performance, as observed in our experiments. This explains why ℓ_∞-norm MKL is the best prediction model in this experiment.

6.5 Execution Time

In this section we demonstrate the efficiency of our implementations of non-sparse MKL. We experiment on the MNIST data set,[15] where the task is to separate odd vs. even digits. The digits in this $n = 60,000$-elemental data set are of size 28x28 leading to $d = 784$ dimensional examples. We compare our analytical solver for non-sparse MKL (Section 4.3.1–4.3.2) with the state-of-the art

15. This data set is available from `http://yann.lecun.com/exdb/mnist/`.

	AUC $\pm$ stderr	
EXP	71.69 ± 1.1	(69.3 ± 1.9)
LOC	58.35 ± 0.7	(56.0 ± 3.3)
PHY	73.35 ± 1.9	(67.8 ± 2.1)
INT (∞-norm MKL)	$\mathbf{82.94 \pm 1.1}$	$(\mathbf{82.1 \pm 2.2})$
1-norm MKL	75.08 ± 1.4	
$4/3$-norm MKL	78.14 ± 1.6	
2-norm MKL	80.12 ± 1.8	
4-norm MKL	81.58 ± 1.9	
8-norm MKL	81.99 ± 2.0	
10-norm MKL	$\mathbf{82.02 \pm 2.0}$	
Recombined and product kernels		
1-norm MKL	79.05 ± 0.5	
$4/3$-norm MKL	80.92 ± 0.6	
2-norm MKL	81.95 ± 0.6	
4-norm MKL	$\mathbf{83.13 \pm 0.6}$	

Table 2: Results for the reconstruction of a metabolic gene network. Results by Bleakley et al. (2007) for single kernel SVMs are shown in brackets.

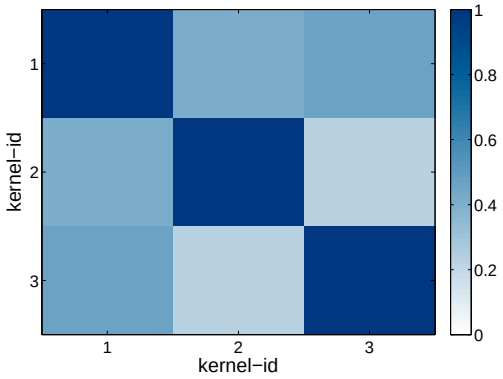

Figure 6: Pairwise alignments of the kernel matrices are shown for the metabolic gene network experiment. From left to right, the ordering of the kernel matrices is EXP, LOC, and PHY. One can see that all kernel matrices are equally correlated. Generally, the alignments are relatively low, suggesting that combining all kernels with equal weights is beneficial.

for ℓ_1-norm MKL, namely SimpleMKL[16] (Rakotomamonjy et al., 2008), HessianMKL[17] (Chapelle and Rakotomamonjy, 2008), SILP-based wrapper, and SILP-based chunking optimization (Sonnenburg et al., 2006a). We also experiment with the analytical method for $p = 1$, although convergence

16. We obtained an implementation from `http://asi.insa-rouen.fr/enseignants/~arakotom/code/`.
17. We obtained an implementation from `http://olivier.chapelle.cc/ams/hessmkl.tgz`.

is only guaranteed by our Theorem 4 for $p > 1$. We also compare to the semi-infinite program (SIP) approach to ℓ_p-norm MKL presented in Kloft et al. (2009a).[18] In addition, we solve standard SVMs[19] using the unweighted-sum kernel (ℓ_∞-norm MKL) as baseline.

We experiment with MKL using precomputed kernels (excluding the kernel computation time from the timings) and MKL based on on-the-fly computed kernel matrices measuring training time *including kernel computations*. Naturally, runtimes of on-the-fly methods should be expected to be higher than the ones of the precomputed counterparts. 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 and HessianMKL, we set their stopping criteria to the relative duality gap of their ℓ_1-norm SILP counterpart. SVM trade-off parameters are set to $C = 1$ for all methods.

6.5.1 SCALABILITY OF THE ALGORITHMS W.R.T. SAMPLE SIZE

Figure 7 (top) displays the results for varying sample sizes and 50 precomputed or on-the-fly computed Gaussian kernels with bandwidths $2\sigma^2 \in 1.2^{0,\ldots,49}$. Error bars indicate standard error over 5 repetitions. As expected, the SVM with the unweighted-sum kernel using precomputed kernel matrices is the fastest method. The classical MKL wrapper based methods, SimpleMKL and the SILP wrapper, are the slowest; they are even slower than methods that compute kernels on-the-fly. Note that the on-the-fly methods naturally have higher runtimes because they do not profit from precomputed kernel matrices.

Notably, when considering 50 kernel matrices of size 8,000 times 8,000 (memory requirements about 24GB for double precision numbers), SimpleMKL is the slowest method: it is more than 120 times slower than the ℓ_1-norm SILP solver from Sonnenburg et al. (2006a). This is because SimpleMKL suffers from having to train an SVM to full precision for each gradient evaluation. In contrast, kernel caching and interleaved optimization still allow to train our algorithm on kernel matrices of size 20000×20000, which would usually not completely fit into memory since they require about 149GB.

Non-sparse MKL scales similarly as ℓ_1-norm SILP for both optimization strategies, the analytic optimization and the sequence of SIPs. Naturally, the generalized SIPs are slightly slower than the SILP variant, since they solve an additional series of Taylor expansions within each θ-step. HessianMKL ranks in between on-the-fly and non-sparse interleaved methods.

6.5.2 SCALABILITY OF THE ALGORITHMS W.R.T. THE NUMBER OF KERNELS

Figure 7 (bottom) shows the results for varying the number of precomputed and on-the-fly computed RBF kernels for a fixed sample size of 1000. The bandwidths of the kernels are scaled such that for M kernels $2\sigma^2 \in 1.2^{0,\ldots,M-1}$. As expected, the SVM with the unweighted-sum kernel is hardly affected by this setup, taking an essentially constant training time. The ℓ_1-norm MKL by Sonnenburg et al. (2006a) handles the increasing number of kernels best and is the fastest MKL method. Non-sparse approaches to MKL show reasonable run-times, being just slightly slower. Thereby the analytical methods are somewhat faster than the SIP approaches. The sparse analytical method

18. The Newton method presented in the same paper performed similarly most of the time but sometimes had convergence problems, especially when $p \approx 1$ and thus was excluded from the presentation.
19. We use SVMlight as SVM-solver.

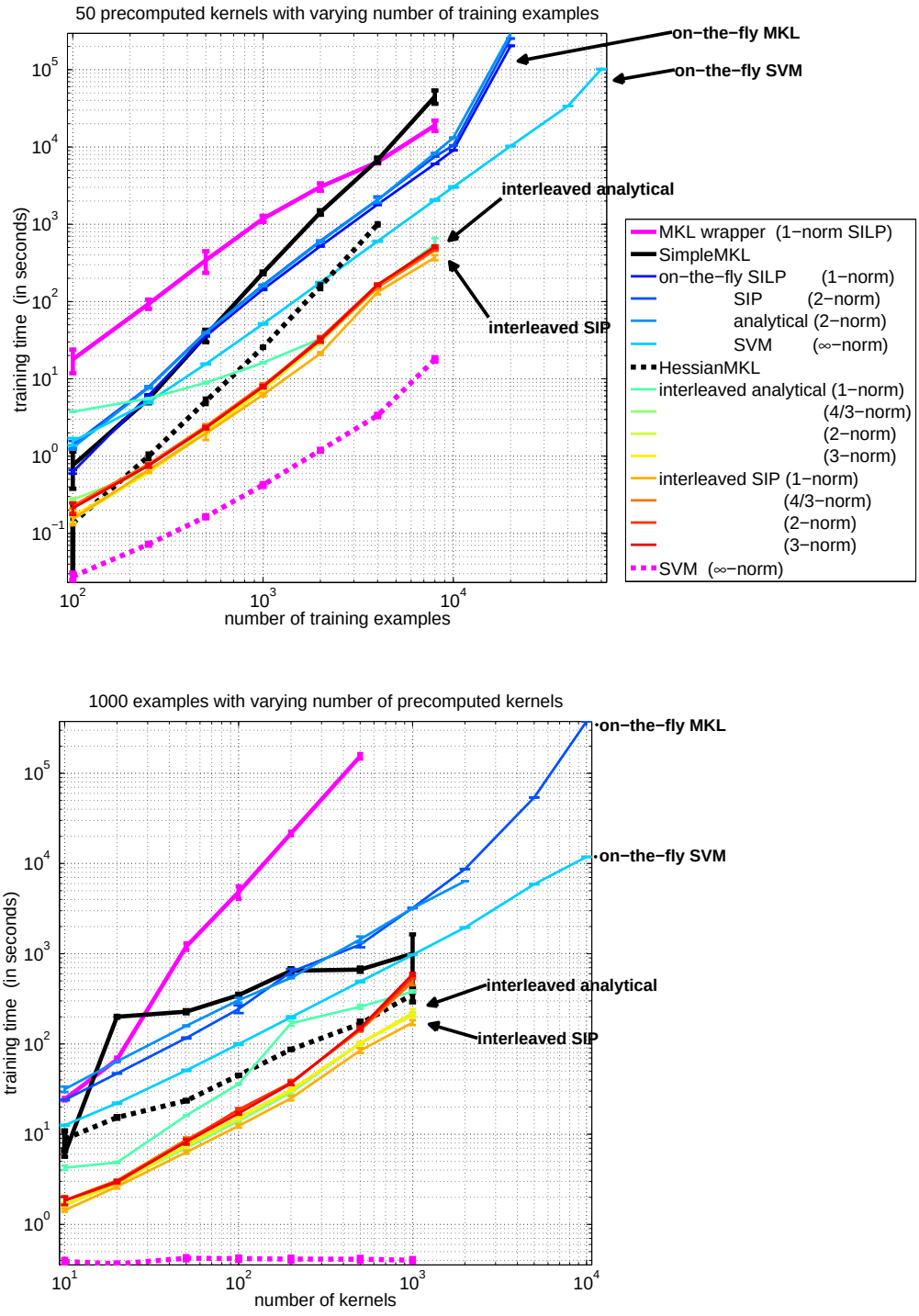

Figure 7: Results of the runtime experiment. Top: Training using fixed number of 50 kernels varying training set size. Bottom: For 1000 examples and varying numbers of kernels. Notice the tiny error bars and that these are log-log plots. The legend is sorted correspondingly to the curves from top to bottom.

performs worse than its non-sparse counterpart; this might be related to the fact that convergence of the analytical method is only guaranteed for $p > 1$. The wrapper methods again perform worst.

However, in contrast to the previous experiment, SimpleMKL becomes more efficient with increasing number of kernels. We conjecture that this is in part owed to the sparsity of the best solution, which accommodates the l_1-norm model of SimpleMKL. But the capacity of SimpleMKL remains limited due to memory restrictions of the hardware. For example, for storing 1,000 kernel matrices for 1,000 data points, about 7.4GB of memory are required. On the other hand, our interleaved optimizers which allow for effective caching can easily cope with 10,000 kernels of the same size (74GB). HessianMKL is considerably faster than SimpleMKL but slower than the non-sparse interleaved methods and the SILP. Similar to SimpleMKL, it becomes more efficient with increasing number of kernels but eventually runs out of memory.

Overall, our proposed interleaved analytic and cutting plane based optimization strategies achieve a speedup of up to one and two orders of magnitude over HessianMKL and SimpleMKL, respectively. Using efficient kernel caching, they allow for truely large-scale multiple kernel learning well beyond the limits imposed by having to precompute and store the complete kernel matrices. Finally, we note that performing MKL with 1,000 precomputed kernel matrices of size 1,000 times 1,000 requires less than 3 minutes for the SILP. This suggests that it focussing future research efforts on improving the accuracy of MKL models may pay off more than further accelerating the optimization algorithm.

7. Conclusion

In the past years, multiple kernel learning research has focused on *accelerating* algorithms for learning convex combinations of kernels. Unfortunately, empirical evidence often showed that sparse MKL-optimized kernel combinations rarely help in practice. By proposing *ℓ_p-norm multiple kernel learning*, conceiving an optimization scheme of unprecedented efficiency, and providing a really efficient implementation (`http://doc.ml.tu-berlin.de/nonsparse_mkl/`), this paper finally makes large-scale MKL practical and profitable.

These advances are founded on our novel general multiple kernel learning framework that subsumes many seemingly different approaches and provides a unifying view and new insights on MKL. In a theoretical analysis, we derived sharp generalization bounds showing that in a non-sparse scenario ℓ_p-norm MKL yields strictly better bounds than ℓ_1-norm MKL and vice versa. However, the difference between the ℓ_p and ℓ_1-norm bounds might not be sufficiently large to completely explain our empirical results. Using the local Rademacher complexity for ℓ_p-norm MKL, one may obtain even tighter bounds, for which the results in Section 5 may serve as a starting point.

In an extensive empirical evaluation, we showed that ℓ_p-norm MKL can significantly improve classification accuracies on diverse and relevant real-world data sets from bioinformatics. Using artificial data, we provided insights by connecting the ℓ_p-norm with the size of the true sparsity pattern. A related—and obtruding!—question is whether the optimality of the parameter p can retrospectively be explained or, more profitably, even be estimated in advance. Clearly, cross-validation based model selection over the choice of p will inevitably tell us which cases call for sparse or non-sparse models. The analyses of our real-world applications suggests that both the correlation amongst the kernels with each other and their correlation with the target (i.e., the amount of discriminative information that they carry) play a role in the distinction of sparse from non-sparse scenarios.

We not only provide a thorough theoretical and empirical analysis, but also contribute an efficient and freely available implementation useful for large-scale real-world applications.

Finally, we would like to note that it may be worthwhile to rethink the current strong preference for sparse models in the scientific community. For example, already weak connectivity in a causal graphical model may be sufficient for all variables to be required for optimal predictions, and even the prevalence of sparsity in causal flows is being questioned (e.g., for the social sciences Gelman, 2010 argues that "There are (almost) no true zeros"). A main reason for favoring sparsity may be the presumed interpretability of sparse models. However, in general sparse MKL solutions are sensitive to kernel normalization, and in particular in the presence of strongly correlated kernels the selection of kernels may be somewhat arbitrary. This puts the interpretation of sparsity patterns in doubt, and it may be more honest to focus on predictive accuracy. In this respect we demonstrate that non-sparse models may improve quite impressively over sparse ones.

Acknowledgments

The authors wish to thank Vojtech Franc, Peter Gehler, Pavel Laskov for stimulating discussions; and Yoann Fabre, Chris Hinrichs, and Klaus-Robert Müller for helpful comments on the manuscript. We thank Motoaki Kawanabe for a valuable suggestion that improved the design of the toy experiment and Gilles Blanchard for a comment that lead to a tighter generalization bound. We acknowledge Peter L. Bartlett and Ulrich Rückert for contributions to parts of an earlier version of the theoretical analysis that appeared at ECML 2010. We thank the anonymous reviewers for comments and suggestions that helped to improve the manuscript. This work was supported in part by the German Bundesministerium für Bildung und Forschung (BMBF) under the project REMIND (FKZ 01-IS07007A), and by the FP7-ICT program of the European Community, under the PASCAL2 Network of Excellence, ICT-216886. Sören Sonnenburg acknowledges financial support by the German Research Foundation (DFG) under the grant MU 987/6-1 and RA 1894/1-1, and Marius Kloft acknowledges a scholarship by the German Academic Exchange Service (DAAD).

Appendix A. Switching Between Tikhonov and Ivanov Regularization

In this appendix, we show a useful result that justifies switching from Tikhonov to Ivanov regularization and vice versa, if the bound on the regularizing constraint is tight. It is the key ingredient of the proof of Theorem 1. We state the result for arbitrary convex functions, so that it can be applied beyond the multiple kernel learning framework of this paper.

Proposition 12 *Let $D \subset \mathbb{R}^d$ be a convex set, let $f, g : D \to \mathbb{R}$ be convex functions. Consider the convex optimization tasks*

$$\min_{x \in D} \quad f(x) + \sigma g(x), \tag{29}$$

$$\min_{x \in D : g(x) \leq \tau} \quad f(x). \tag{30}$$

Assume that the minima exist and that a constraint qualification holds in (30)*, which gives rise to strong duality, for example, that Slater's condition is satisfied. Furthermore assume that the*

constraint is active at the optimal point, that is,

$$\inf_{x\in D} f(x) \;<\; \inf_{x\in D: g(x)\leq \tau} f(x). \tag{31}$$

Then we have that for each $\sigma > 0$ *there exists* $\tau > 0$*—and vice versa—such that OP* (29) *is equivalent to OP* (30)*, that is, each optimal solution of one is an optimal solution of the other, and vice versa.*

Proof
(a). Let be $\sigma > 0$ and x^* be the optimal of (29). We have to show that there exists a $\tau > 0$ such that x^* is optimal in (30). We set $\tau = g(x^*)$. Suppose x^* is not optimal in (30), that is, it exists $\tilde{x} \in D : g(\tilde{x}) \leq \tau$ such that $f(\tilde{x}) < f(x^*)$. Then we have

$$f(\tilde{x}) + \sigma g(\tilde{x}) < f(x^*) + \sigma\tau,$$

which by $\tau = g(x^*)$ translates to

$$f(\tilde{x}) + \sigma g(\tilde{x}) < f(x^*) + \sigma g(x^*).$$

This contradics the optimality of x^* in (29), and hence shows that x^* is optimal in (30), which was to be shown.
(b). Vice versa, let $\tau > 0$ be x^* optimal in (30). The Lagrangian of (30) is given by

$$\mathcal{L}(\sigma) = f(x) + \sigma\,(g(x) - \tau)\,, \quad \sigma \geq 0.$$

By strong duality x^* is optimal in the saddle point problem

$$\sigma^* := \operatorname*{argmax}_{\sigma\geq 0} \min_{x\in D} \quad f(x) + \sigma\,(g(x) - \tau)\,,$$

and by the strong max-min property (cf. Boyd and Vandenberghe, 2004, p. 238) we may exchange the order of maximization and minimization. Hence x^* is optimal in

$$\min_{x\in D} \quad f(x) + \sigma^*\,(g(x) - \tau)\,. \tag{32}$$

Removing the constant term $-\sigma^*\tau$, and setting $\sigma = \sigma^*$, we have that x^* is optimal in (29), which was to be shown. Moreover by (31) we have that

$$x^* \neq \operatorname*{argmin}_{x\in D} f(x),$$

and hence we see from Equation (32) that $\sigma^* > 0$, which completes the proof of the proposition. ■

References

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

J. Aflalo, A. Ben-Tal, C. Bhattacharyya, J. Saketha Nath, and S. Raman. Variable sparsity kernel learning—algorithms and applications. *Journal of Machine Learning Research*, 2011. To appear.

A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

F. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 105–112, 2009.

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.

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.

P.L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, November 2002.

D.P. Bertsekas. *Nonlinear Programming, Second Edition*. Athena Scientific, Belmont, MA, 1999.

K. Bleakley, G. Biau, and J.-P. Vert. Supervised reconstruction of biological networks with local models. *Bioinformatics*, 23:i57–i65, 2007.

O. Bousquet and D.J.L. Herrmann. On the complexity of learning the kernel matrix. In *Advances in Neural Information Processing Systems*, 2002.

O. Bousquet, S. Boucheron, and G. Lugosi. Introduction to statistical learning theory. In Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch, editors, *Advanced Lectures on Machine Learning*, volume 3176 of *Lecture Notes in Computer Science*, pages 169–207. Springer Berlin / Heidelberg, 2004.

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

O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 2006.

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.

O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1):131–159, 2002.

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. URL `http://www.cs.nyu.edu/learning_kernels`.

C. Cortes, M. Mohri, and A. Rostamizadeh. L2 regularization for learning kernels. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, 2009a.

C. Cortes, M. Mohri, and A. Rostamizadeh. Learning non-linear combinations of kernels. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 396–404, 2009b.

C. Cortes, M. Mohri, and A. Rostamizadeh. Generalization bounds for learning kernels. In *Proceedings, 27th ICML*, 2010a.

C. Cortes, M. Mohri, and A. Rostamizadeh. Two-stage learning kernel algorithms. In *Proceedings of the 27th Conference on Machine Learning (ICML 2010)*, 2010b.

N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor. On kernel-target alignment. In *Advances in Neural Information Processing Systems*, 2002.

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Number 31 in Applications of Mathematics. Springer, New York, 1996.

R. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

R.-E. Fan, P.-H. Chen, and C.-J. Lin. Working set selection using the second order information for training support vector machines. *Journal of Machine Learning Research*, 6:1889–1918, 2005.

V. Franc and S. Sonnenburg. OCAS optimized cutting plane algorithm for support vector machines. In *Proceedings of the 25nd International Machine Learning Conference*. ACM Press, 2008.

P. V. Gehler and S. Nowozin. Infinite kernel learning. In *Proceedings of the NIPS 2008 Workshop on Kernel Learning: Automatic Selection of Optimal Kernels*, 2008.

A. Gelman. Causality and statistical learning. *American Journal of Sociology*, 0, 2010.

G.H. Golub and C.F. van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, London, 3rd edition, 1996.

M. Gönen and E. Alpaydin. Localized multiple kernel learning. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 352–359, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: http://doi.acm.org/10.1145/1390156.1390201.

V.K. Ivanov, V.V. Vasin, and V.P. Tanana. *Theory of Linear Ill-Posed Problems and its application*. VSP, Zeist, 2002.

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

T. Joachims. Making large–scale SVM learning practical. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 169–184, Cambridge, MA, 1999. MIT Press.

S. M. Kakade, S. Shalev-Shwartz, and A. Tewari. Applications of strong convexity–strong smoothness duality to learning with matrices. *CoRR*, abs/0910.0610, 2009.

M. Kanehisa, S. Goto, S. Kawashima, Y. Okuno, and M. Hattori. The KEGG resource for deciphering the genome. *Nucleic Acids Res*, 32:D277–D280, 2004.

G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *J. Math. Anal. Applic.*, 33:82–95, 1971.

M. Kloft, U. Brefeld, P. Laskov, and S. Sonnenburg. Non-sparse multiple kernel learning. In *Proc. of the NIPS Workshop on Kernel Learning: Automatic Selection of Kernels*, dec 2008.

M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Müller, and A. Zien. Efficient and accurate lp-norm multiple kernel learning. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 997–1005. MIT Press, 2009a.

M. Kloft, S. Nakajima, and U. Brefeld. Feature selection for density level-sets. In W. L. Buntine, M. Grobelnik, D. Mladenic, and J. Shawe-Taylor, editors, *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pages 692–704, 2009b.

M. Kloft, U. Rückert, and P. L. Bartlett. A unifying view of multiple kernel learning. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, 2010. To appear. ArXiv preprint: `http://arxiv.org/abs/1005.0437`.

V. Koltchinskii and D. Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 30:1–50, 2002.

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.

D.C. Liu and J. Nocedal. On the limited memory method for large scale optimization. *Mathematical Programming B*, 45(3):503–528, 1989.

P. C. Mahalanobis. On the generalised distance in statistics. In *Proceedings National Institute of Science, India*, volume 2, no. 1, April 1936.

M. Markou and S. Singh. Novelty detection: a review – part 1: statistical approaches. *Signal Processing*, 83:2481–2497, 2003a.

M. Markou and S. Singh. Novelty detection: a review – part 2: neural network based approaches. *Signal Processing*, 83:2499–2521, 2003b.

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

K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Neural Networks*, 12(2):181–201, May 2001.

S. Nash and A. Sofer. *Linear and Nonlinear Programming*. McGraw-Hill, New York, NY, 1996.

J. S. Nath, G. Dinesh, S. Ramanand, C. Bhattacharyya, A. Ben-Tal, and K. R. Ramakrishnan. On the algorithmics and applications of a mixed-norm based kernel learning formulation. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 844–852, 2009.

A. Nemirovski. Prox-method with rate of convergence o(1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15:229–251, 2004.

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.

C. S. Ong, A. J. Smola, and R. C. Williamson. Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, 6:1043–1071, 2005.

S. Özögür-Akyüz and G.W. Weber. Learning with infinitely many kernels via semi-infinite programming. In *Proceedings of Euro Mini Conference on Continuous Optimization and Knowledge Based Technologies*, 2008.

J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 185–208, Cambridge, MA, 1999. MIT Press.

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

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

R. M. Rifkin and R. A. Lippert. Value regularization and Fenchel duality. *J. Mach. Learn. Res.*, 8: 441–479, 2007.

V. Roth and B. Fischer. Improved functional prediction of proteins by learning kernel combinations in multilabel settings. *BMC Bioinformatics*, 8(Suppl 2):S12, 2007. ISSN 1471-2105.

V. Roth and B. Fischer. The group-lasso for generalized linear models: uniqueness of solutions and efficient algorithms. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning (ICML 2008)*, volume 307, pages 848–855. ACM, 2008.

E. Rubinstein. Support vector machines via advanced optimization techniques. Master's thesis, Faculty of Electrical Engineering, Technion, 2005, Nov 2005.

W. Rudin. *Functional Analysis*. McGraw-Hill, 1991.

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

B. Schölkopf, A.J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

B. Schölkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A.J. Smola. Input space vs. feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5): 1000–1017, September 1999.

B. Schölkopf, J. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

S. Sonnenburg, G. Rätsch, and C. Schäfer. Learning interpretable SVMs for biological sequence classification. In *RECOMB 2005, LNBI 3500*, pages 389–407. Springer-Verlag Berlin Heidelberg, 2005.

S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, July 2006a.

S. Sonnenburg, A. Zien, and G. Rätsch. Arts: Accurate recognition of transcription starts in human. *Bioinformatics*, 22(14):e472–e480, 2006b.

S. Sonnenburg, G. Rätsch, S. Henschel, C. Widmer, J. Behr, A. Zien, F. de Bona, A. Binder, C. Gehl, and V. Franc. The SHOGUN Machine Learning Toolbox. *Journal of Machine Learning Research*, 2010.

J. M. Steele. *The Cauchy-Schwarz Master Class: An Introduction to the Art of Mathematical Inequalities*. Cambridge University Press, New York, NY, USA, 2004. ISBN 052154677X.

M. Stone. Cross-validatory choice and assessment of statistical predictors (with discussion). *Journal of the Royal Statistical Society*, B36:111–147, 1974.

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.

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

M. Szafranski, Y. Grandvalet, and A. Rakotomamonjy. Composite kernel learning. *Mach. Learn.*, 79(1-2):73–103, 2010. ISSN 0885-6125. doi: http://dx.doi.org/10.1007/s10994-009-5150-6.

D.M.J. Tax and R.P.W. Duin. Support vector domain description. *Pattern Recognition Letters*, 20 (11–13):1191–1199, 1999.

A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed problems*. W. H. Winston, Washington, 1977.

M. Varma and B. R. Babu. More generality in efficient multiple kernel learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 1065–1072, New York, NY, USA, 2009. ACM.

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.

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.

Z. Xu, R. Jin, H. Yang, I. King, and M. Lyu. Simple and efficient multiple kernel learning by group lasso. In *Proceedings of the 27th Conference on Machine Learning (ICML 2010)*, 2010.

Y. Yamanishi, , J.-P. Vert, and M. Kanehisa. Supervised enzyme network inference from the integration of genomic data and chemical information. *Bioinformatics*, 21:i468–i477, 2005.

Y. Ying, C. Campbell, T. Damoulas, and M. Girolami. Class prediction from disparate biological data sources using an iterative multi-kernel algorithm. In Visakan Kadirkamanathan, Guido Sanguinetti, Mark Girolami, Mahesan Niranjan, and Josselin Noirel, editors, *Pattern Recognition in Bioinformatics*, volume 5780 of *Lecture Notes in Computer Science*, pages 427–438. Springer Berlin / Heidelberg, 2009.

S. Yu, T. Falck, A. Daemen, L.-C. Tranchevent, J. Suykens, B. De Moor, and Y. Moreau. L2-norm multiple kernel learning and its application to biomedical data fusion. *BMC Bioinformatics*, 11 (1):309, 2010. ISSN 1471-2105.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.

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.