Non-Sparse Regularization and Efficient Training with Multiple Kernels

Marius Kloft
Ulf Brefeld
Sören Sonnenburg
Alexander Zien

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2010-21
http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-21.html

February 24, 2010

Acknowledgement

The authors wish to thank Pavel Laskov, Motoaki Kawanabe, Vojtech Franc, Peter Gehler, Gunnar Raetsch, Peter Bartlett and Klaus-Robert Mueller for fruitful discussions and helpful comments. This work was supported in part by the German Bundesministerium fuer Bildung und Forschung (BMBF) under the project REMIND (FKZ 01-IS07007A), by the German Academic Exchange Service, and by the FP7-ICT Programme of the European Community, under the PASCAL2 Network of Excellence, ICT-216886. Soeren Sonnenburg acknowledges financial support by the German Research Foundation (DFG) under the grant MU 987/6-1 and RA 1894/1-1.

Non-Sparse Regularization and Efficient Training with Multiple Kernels

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@TUEBINGEN.MPG.DE
Friedrich Miescher Laboratory
Max Planck Society
Spemannstr. 39, 72076 Tübingen, Germany

Alexander Zien ZIEN@LIFEBIOSYSTEMS.COM
LIFE Biosystems GmbH
Poststraße 34
69115 Heidelberg, Germany

Abstract

Learning linear combinations of multiple kernels is an appealing strategy when the right choice of features is unknown. Previous approaches to multiple kernel learning (MKL) promote sparse kernel combinations to support interpretability and scalability. Unfortunately, this ℓ_1-norm MKL is rarely observed to outperform trivial baselines in practical applications. To allow for robust kernel mixtures, we generalize 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, like ℓ_p-norms with $p > 1$. Empirically, we demonstrate that the interleaved optimization strategies are much faster compared to the commonly used wrapper approaches. An experiment on controlled artificial data experiment sheds light on the appropriateness of sparse, non-sparse and ℓ_∞ MKL in various scenarios. Application of ℓ_p-norm MKL to three hard real-world problems from computational biology show that non-sparse MKL achieves accuracies that go beyond the state-of-the-art. We conclude that our improvements finally made MKL fit for deployment to practical applications: MKL now has a good chance of improving the accuracy (over a plain sum kernel) at an affordable computational cost.

1. Introduction

Kernels allow to decouple machine learning from data. Finding an appropriate data representation via a kernel function immediately opens the door to a vast world of powerful

∗. Also at Machine Learning Group, Technische Universität Berlin, Franklinstr. 28/29, FR 6-9, 10587 Berlin, Germany.

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) 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 (Varma and Babu, 2009) generally results in non-convex optimization problems, Cortes et al. (2009) 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 focused on devising efficient optimization procedures for learning with multiple kernels.

One conceptual milestone for developing MKL into a tool of practical utility is simply to constrain the mixing coefficients $\boldsymbol{\theta}$ to be non-negative: by obviating the complex constraint $K \succeq 0$, this small restriction allows one 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 $\boldsymbol{\theta}$. This immediately lends itself to a convenient implementation by a wrapper approach. These algorithms directly benefit from efficient SVM optimization routines (cf., e.g., 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 multiple kernel learning 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 proposed to interleave the SILP with the SVM training which reduced 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. (2008), 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 mayor families of multiple kernel learning models, characterized either by Ivanov regularization (Ivanov et al., 2002) over the mixing coefficients (Rakotomamonjy et al., 2007; Zien and Ong, 2007), or as Tikhonov regularized optimization problem (Tikhonov and Arsenin, 1977). In the both cases, there may be 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 $\|\boldsymbol{\theta}\|_1 \leq 1$ simplifies derivations and turns the problem into a linearly constrained program. Nevertheless, sparseness is not always beneficial in practice. Sparse MKL is frequently observed to be outperformed by a regular SVM using an unweighted-sum kernel $K = \sum_m K_m$.

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, at least when considering MKL restricted to non-negative mixture coefficients. 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 $\boldsymbol{\theta}$. 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 generalized dual and show that a variety of methods are special cases of our objective. Our detached 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 $\boldsymbol{\theta}$-step; hence rendering expensive semi-infinite and first- or second-order gradient methods unnecessary. By utilizing 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 from `http://www.shogun-toolbox.org/`.

Our claims are backed up by experiments on artificial data and on a couple of real world data sets representing diverse, relevant and challenging problems from the application domain bioinformatics. The artificial data enables us to investigate the relationship between properties of the true solution and the optimal choice of kernel mixture regularization. The real world problems include the prediction of the subcellular localization of proteins, the (transcription) starts of genes, and the function of enzymes. 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.

The remainder of this paper is structured as follows. We derive the generalized MKL in Section 2 and discuss relations to existing approaches in Section 3. Section 4 introduces the novel optimization strategy and shows the applicability of existing optimization techniques to our generalized formulation. We report on our empirical results in Section 5. Section 6 concludes.

2. Generalized MKL

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

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

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

We consider linear models of the form

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

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 $\Omega(f) = \frac{1}{2}||\tilde{\boldsymbol{w}}||_2^2$ which allows to kernelize the resulting models and algorithms. We will later make use of kernel functions $K(\boldsymbol{x}, \boldsymbol{x}') = \langle \psi(\boldsymbol{x}), \psi(\boldsymbol{x}') \rangle_{\mathcal{H}}$ to compute inner products in $\mathcal{H}$.

2.2 Convex Risk Minimization with Multiple Kernels

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

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

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

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

$$\inf_{\tilde{\boldsymbol{w}},b,\boldsymbol{\theta}:\boldsymbol{\theta}\geq\boldsymbol{0}} \quad \frac{1}{n}\sum_{i=1}^{n} V\left(\sum_{m=1}^{M} \sqrt{\theta_m}\langle \tilde{\boldsymbol{w}}_m, \psi_m(\boldsymbol{x}_i)\rangle_{\mathcal{H}_m} + b,\ y_i\right) + \frac{\lambda}{2}\sum_{m=1}^{M} ||\tilde{w}_m||_{\mathcal{H}_m}^2 + \tilde{\mu}\tilde{\Omega}[\boldsymbol{\theta}], \tag{3}$$

for $\tilde{\mu} > 0$. Note that the objective value of Eq. (3) is an upper bound on the training error. Previous approaches to multiple kernel learning employ regularizers of the form $\tilde{\Omega}(\boldsymbol{\theta}) = ||\boldsymbol{\theta}||_1$ to promote sparse kernel mixtures. By contrast, we propose to use convex regularizers of the form $\tilde{\Omega}(\boldsymbol{\theta}) = ||\boldsymbol{\theta}||^2$, 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 $\sqrt{\theta_m}\tilde{\boldsymbol{w}}_m$ product in the loss term of Eq. (3) is not inherent and can be resolved by substituting $\boldsymbol{w}_m \leftarrow \sqrt{\theta_m}\tilde{\boldsymbol{w}}_m$. Furthermore, the regularization parameter and the sample size can be decoupled by introducing $\tilde{C} = \frac{1}{n\lambda}$ (and adjusting $\mu \leftarrow \frac{\tilde{\mu}}{\lambda}$) which has favorable scaling properties in practice. We obtain the following convex optimization problem (Boyd and Vandenberghe, 2004) that has also been considered by (Varma and Ray, 2007) for hinge loss and an ℓ_1-norm regularizer

$$\inf_{\boldsymbol{w},b,\boldsymbol{\theta}:\boldsymbol{\theta}\geq\boldsymbol{0}} \quad \tilde{C}\sum_{i=1}^{n} V\left(\sum_{m=1}^{M} \langle \boldsymbol{w}_m, \psi_m(\boldsymbol{x}_i)\rangle_{\mathcal{H}_m} + b,\ y_i\right) + \frac{1}{2}\sum_{m=1}^{M} \frac{||\boldsymbol{w}_m||_{\mathcal{H}_m}^2}{\theta_m} + \mu||\boldsymbol{\theta}||^2, \tag{4}$$

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 $||\boldsymbol{\theta}||_1 \leq 1$ and incorporate the latter as an additional constraint into the optimization problem. For $C > 0$, they arrive at the following problem which is the primary object of investigation in this paper.

Primal MKL Optimization Problem

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

Our first contribution shows that despite the additional regularization parameter the Tikhonov regularization in (4) and the Ivanov regularization in Optimization Problem (P) are equivalent, in the sense that they yield the same binary classification function.

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

For the proof we need Prop. 8, which justifies switching from Ivanov to Tikhonov regularization, and back, if the regularizer is tight. We refer to Appendix A for formulation and proof of the proposition.

Proof of Theorem 1 Let be $(\tilde{C}, \mu) > 0$. In order to apply Prop. 8 to (4), we start by showing that condition (35) in Prop. 8 is satisfied, i.e., that the regularizer is tight.

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

$$||\boldsymbol{\theta}||^2 \leq 1, \tag{5}$$

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

$$\frac{||\boldsymbol{w}_m^*||_2^2}{\theta_m^*} = 0 \ \forall\ m. \tag{6}$$

Since all norms on $\mathbb{R}^M$ are equivalent (cf., e.g., Rudin (1991)), there exists a $L < \infty$ such that $||\boldsymbol{\theta}^*||_\infty \leq L||\boldsymbol{\theta}^*||$. In particular, we have $||\boldsymbol{\theta}^*||_\infty < \infty$, from which we conclude by (6), that $\boldsymbol{w}_m = 0$ holds for all m, which contradicts our assumption.

Hence, Prop. 8 can be applied and which yields that (4) is equivalent to

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

for some $\tau > 0$. Consider the optimal solution $(\boldsymbol{w}^\star, b^\star, \boldsymbol{\theta}^\star)$ 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 $(\boldsymbol{w}^\star, b^\star, \lambda^{1/2}\boldsymbol{\theta}^\star)$ 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 (P), which was to be shown. ■

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

2.3 Convex 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 (P) by expanding the decision values into slack variables as follows

$$\begin{aligned} \inf_{\boldsymbol{w},b,\boldsymbol{t},\boldsymbol{\theta}:\boldsymbol{\theta}\geq\mathbf{0}} \quad & C\sum_{i=1}^{n} V\left(t_i,\ y_i\right) + \frac{1}{2}\sum_{m=1}^{M}\frac{||\boldsymbol{w}_m||^2_{\mathcal{H}_m}}{\theta_m} \\ \text{s.t.} \quad & \forall i:\ \sum_{m=1}^{M}\langle \boldsymbol{w}_m, \psi_m(\boldsymbol{x}_i)\rangle_{\mathcal{H}_m} + b = t_i\ ; \quad ||\boldsymbol{\theta}||^2 \leq 1, \end{aligned} \tag{7}$$

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 multipliers $\boldsymbol{\alpha} \in \mathbb{R}^n$ and $\beta \in \mathbb{R}_+$. [1] The Lagrangian saddle point problem is then given by

$$\begin{aligned} \sup_{\boldsymbol{\alpha},\beta:\beta\geq 0}\ \inf_{\boldsymbol{w},b,\boldsymbol{t},\boldsymbol{\theta}\geq 0} \quad & C\sum_{i=1}^{n} V\left(t_i,\ y_i\right) + \frac{1}{2}\sum_{m=1}^{M}\frac{||\boldsymbol{w}_m||^2_{\mathcal{H}_m}}{\theta_m} \\ & - \sum_{i=1}^{n}\alpha_i\left(\sum_{m=1}^{M}\langle \boldsymbol{w}_m, \psi_m(\boldsymbol{x}_i)\rangle_{\mathcal{H}_m} + b - t_i\right) + \beta\left(\frac{1}{2}||\boldsymbol{\theta}||^2 - \frac{1}{2}\right). \end{aligned} \tag{8}$$

1. Note that $\boldsymbol{\alpha}$ is variable over the whole range of $\mathbb{R}^n$ since it is incorporates an equality constraint.

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

$$\mathbf{1}^\top \alpha = 0; \tag{9a}$$

$$\forall\, m = 1, \cdots M: \quad \boldsymbol{w}_m = \theta_m \sum_{i=1}^{n} \alpha_i \psi_m(\boldsymbol{x}_i)\ . \tag{9b}$$

Resubstituting the above equations yields

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

which can also be written in terms of unconstrained $\boldsymbol{\theta}$ because, without loss of generality, a supremum with respect to $\boldsymbol{\theta}$ is trivially attained for arbitrary non-negative $\boldsymbol{\theta}\geq 0$. We arrive at

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

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

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

where $h^*(\boldsymbol{x})=\sup_{\boldsymbol{u}}\boldsymbol{x}^\top\boldsymbol{u}-h(\boldsymbol{u})$ denotes the Fenchel-Legendre conjugate of a function h and $||\cdot||_*$ denotes the *dual norm*, i.e., 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*. Eq. (10) now has to be maximized with respect to the dual variables $\boldsymbol{\alpha},\beta$, subject to $\mathbf{1}^\top\boldsymbol{\alpha}=0$ 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

$$\beta=\left\|\frac{1}{2}\left(\boldsymbol{\alpha}^\top K_m\boldsymbol{\alpha}\right)_{m=1}^{M}\right\|_*. \tag{11}$$

Obviously, at optimality, we always have $\beta\geq 0$. We thus discard the corresponding constraint from the optimization problem and plugging Eq. (11) into Eq. (10) results in the following *dual* optimization problem which now solely depends on $\boldsymbol{\alpha}$:

Dual MKL Optimization Problem

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

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

The above dual generalizes multiple kernel learning to arbitrary convex loss functions and norms. Note that if the loss function is continuous 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 (D) is a composition of two terms, firstly, the left hand side term, which depends on the conjugate loss function V^*, and, secondly, 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 nice 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 $(V^*, ||\cdot||_*)$. In the next section, this is illustrated by means of well-known loss functions and regularizers.

3. Instantiations of the Model

In this section we show that existing MKL-based learners are subsumed by the generalized formulation in Optimization Problem (D).

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 RRM 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). Hence, for each i the term $V^*\left(-\frac{\alpha_i}{C},\ y_i\right)$ of the generalized dual, i.e., Optimization Problem (D), 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 (D) translates to

$$\max_{\boldsymbol{\alpha}} \quad \mathbf{1}^\top \boldsymbol{\alpha} - \frac{1}{2} \left\| \left(\boldsymbol{\alpha}^\top Y K_m Y \boldsymbol{\alpha} \right)_{m=1}^{M} \right\|_*, \quad \text{s.t.} \quad \boldsymbol{y}^\top \boldsymbol{\alpha} = 0 \ \text{ and } \ \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}, \tag{12}$$

where we denote $Y = \text{diag}(y)$. The primal ℓ_∞-norm penalty $||\boldsymbol{\theta}||_\infty$ is dual to $||\boldsymbol{\theta}||_1$, hence, via the identity $||\cdot||_* = ||\cdot||_1$ the right hand side of the last equation translates to $\sum_{m=1}^{M} \boldsymbol{\alpha}^\top Y K_m Y \boldsymbol{\alpha}$. Combined with (12) this leads to the dual

$$\sup_{\boldsymbol{\alpha}} \quad \mathbf{1}^\top \boldsymbol{\alpha} - \frac{1}{2} \sum_{m=1}^{M} \boldsymbol{\alpha}^\top Y K_m Y \boldsymbol{\alpha}, \quad \text{s.t.} \quad \boldsymbol{y}^\top \boldsymbol{\alpha} = 0 \ \text{ and } \ \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1},$$

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

$$\Omega = ||\theta||_1. \tag{13}$$

This so-called ℓ_1-norm regularizers are specific instances of *sparsity-inducing* regularizers. The obtained kernel mixtures are often *sparse* 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 (13) is $||\cdot||_\infty$. Recalling the definition of an ℓ_p-norm, the right hand side of Optimization Problem (D) translates to $\max_{m\in\{1,\ldots,M\}} \boldsymbol{\alpha}^\top Y K_m Y \boldsymbol{\alpha}$. The maximum can subsequently be expanded into slack variables ξ_i, resulting in

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

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

3.3 ℓ_p-Norm MKL

The generalized MKL also allows for robust kernel mixtures by employing an ℓ_p-norm constraint with $p > 1$, rather than an ℓ_1-norm constraint, on the mixing coefficients (Kloft et al., 2009a). The following identity holds

$$\left(\frac{1}{2}||\cdot||_p^2\right)^* = \frac{1}{2}||\cdot||_q^2, \quad \text{where} \ \ \frac{1}{p}+\frac{1}{q}=1,$$

and we obtain for the dual norm of the ℓ_p-norm: $||\cdot||_* = ||\cdot||_q$. This leads to the dual problem

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

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

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

It is thereby worth mentioning that the optimality conditions yield the proportionality,

$$\theta_m^* \sim (\boldsymbol{\alpha}^* K_m \boldsymbol{\alpha}^*)^{\frac{2}{p-1}},$$

as we will show in Sect. 4.1.

3.4 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,

$$\min_{\boldsymbol{w},b} \quad \frac{C}{2}\sum_{i=1}^n\left(y_i - \sum_{m=1}^M \langle \boldsymbol{w}_m, \psi_m(\boldsymbol{x}_i)\rangle_{\mathcal{H}_m}\right)^2 + \frac{1}{2}\sum_{m=1}^M ||\boldsymbol{w}_m||_{\mathcal{H}_m}. \tag{14}$$

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 Eq. (14) can be equivalently expressed as (Micchelli and Pontil, 2005a)

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

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

$$\max_{\boldsymbol{\alpha}} \quad \boldsymbol{y}^\top\boldsymbol{\alpha} - \frac{1}{2C}||\boldsymbol{\alpha}||_2^2 - \frac{1}{2}\left\|\left(\boldsymbol{\alpha}^\top Y K_m Y \boldsymbol{\alpha}\right)_{m=1}^{M}\right\|_\infty, \tag{15}$$

which can be expanded into the following QCQP

$$\begin{aligned} \sup_{\boldsymbol{\alpha},\xi} \quad & \boldsymbol{y}^\top\boldsymbol{\alpha} - \frac{1}{2C}||\boldsymbol{\alpha}||_2^2 - \xi_i \\ \text{s.t.} \quad & \forall m: \quad \frac{1}{2}\boldsymbol{\alpha}^\top Y K_m Y \boldsymbol{\alpha} \leq \xi_m. \end{aligned} \tag{16}$$

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 a smooth and unconstrained variant based on ℓ_p-norms ($p > 1$), given by

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

which can be solved very efficiently by limited memory quasi-Newton descent methods (Liu and Nocedal, 1989).

3.5 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) have been extended to MKL settings by Sonnenburg et al. (2006a) and Kloft et al. (2008), respectively. One-class MKL can be cast into our framework by employing loss functions of the form $V(t) = \max(0, 1-t)$. This gives rise to the primal

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

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

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

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

3.6 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 have given an estimate of the interactions within the set of kernels considered , e.g. in the form of an $M \times M$ matrix E. Alternatively, it might be known in advance that a subset of the employed kernels is inferior to the remaining kernels; for instance, such knowledge could result from previous experiments in the considered application field. Those scenarios can be easily handled within our framework by considering non-isotropic regularizers of the form

$$||\boldsymbol{\theta}||_E = \sqrt{\boldsymbol{\theta}^\top E \boldsymbol{\theta}} \quad \text{with} \quad E \succ 0.$$

The dual norm is again defined via $\frac{1}{2}||\cdot||_*^2 := \left(\frac{1}{2}||\cdot||_E^2\right)^*$ and the following easily-to-verify identity,

$$\left(\frac{1}{2}||\cdot||_E^2\right)^* = \frac{1}{2}||\cdot||_F^2,$$

with matrix inverse $F = E^{-1}$, leads to the dual,

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

which is the desired non-isotropic MKL problem.

4. Efficient Optimization Strategies

The dual as given in Optimization Problem (D) does not lend itself to efficient large-scale optimization in a straight-forward fashion, for instance by direct application of standard appoaches like gradient descent. Instead, it is beneficial to exploit the structure of the MKL cost function by alternating between optimizing w.r.t. to the mixings $\boldsymbol{\theta}$ and w.r.t. to the remaining variables. Most recent MKL solvers (e.g., Rakotomamonjy et al., 2008; Xu et al., 2009; Varma and Babu, 2009) do so by setting up a two-layer optimization procedure: a master problem, which is parameterized only by $\boldsymbol{\theta}$ and independent of $\boldsymbol{\theta}$, 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 convential, efficient SVM optimizers can be recycled. Consequently these two-layer procedures are commonly implemented as *wrapper* approaches. Albeit appearing advantageous, wrapper methods suffer from a few 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 $\boldsymbol{\theta}$ 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, etc. Therefore

all optimization approaches presented in this paper implement a true decomposition of 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 algorithms are embedded into the large-scale framework of Sonnenburg et al. (2006a) and extend them to optimization of non-sparse kernel mixtures induced by an ℓ_p-norm penalty. Our first strategy alternates between minimizing the primal problem (7) w.r.t. $\boldsymbol{\theta}$ with incomplete optimization w.r.t. all other variables which, however, is performed in terms of the dual variables $\boldsymbol{\alpha}$. For the second strategy, we devise a convex semi-infinite program (SIP), which we solve by column generation with nested sequential quadratically constrained linear programming (SQCLP). In both cases, optimization w.r.t. $\boldsymbol{\alpha}$ is performed by chunking optimization with minor iterations. The first, "direct" approach can be applied without a common purpose QCQP solver. We show convergence of both algorithms: for the "direct" algorithm in Prop. 5 and convergence of the SQCLP in Prop. 6. All algorithms are implemented in the SHOGUN machine learning toolbox, which is freely available from `http://www.shogun-toolbox.org/`.

4.1 An Analytical Method

In this section we present a simple and efficient optimization strategy for multiple kernel learning. To derive the new algorithm, we first revisit the primal problem, i.e.

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

In order to obtain an efficient optimization strategy, we divide the variables in the above OP into two groups, $(\boldsymbol{w}, b)$ on one hand and $\boldsymbol{\theta}$ on the other. In the following we will derive an algorithm, which alternatingly operates on those two groups via block coordinate descent algorithm, also known as the *non-linear Gauss-Seidel method*. Thereby the optimization w.r.t. $\boldsymbol{\theta}$ will be carried out analytically and the $(\boldsymbol{w}, b)$-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 $(\boldsymbol{w}, b)$, the optimal $\boldsymbol{\theta}$ in the primal problem (P) can be calculated analytically. In the subsequent derivations we exemplarily employ non-sparse norms of the form $||\boldsymbol{\theta}||_p = (\sum_{m=1}^{M}\theta_m^p)^{1/p}$, $1 < p < \infty$, but the reasoning—including convergence guarantees—holds for arbitrary continuously differentiable and strictly convex norms[3].

The following proposition gives the an analytic update formula for the θ given fixed remaining variables $(\boldsymbol{w}, b)$ and will become the core of our proposed algorithm.

Proposition 2 *Let V be a convex loss function, be $p > 1$. Given fixed $(\boldsymbol{w}, b)$, the optimal solution of Optimization Problem (P) is attained for*

$$\theta_m^* = \frac{||\boldsymbol{w}_m^*||_{\mathcal{H}_m}^{\frac{2}{p+1}}}{\left(\sum_{m'=1}^{M}||\boldsymbol{w}_{m'}^*||_{\mathcal{H}_{m'}}^{\frac{2p}{p+1}}\right)^{1/p}}, \qquad \forall m = 1,\ldots,M. \quad (17)$$

3. Lemma 26 in Micchelli and Pontil (2005b) indicates that the result could even be extended to an infinite number of kernels.

Proof We start the derivation, by equivalently translating Optimization Problem (P) via Theorem 1 into

$$\inf_{\boldsymbol{w},b,\boldsymbol{\theta}:\boldsymbol{\theta}\geq\mathbf{0}} \quad \tilde{C}\sum_{i=1}^{n} V\left(\sum_{m=1}^{M}\langle \boldsymbol{w}_m, \psi_m(\boldsymbol{x}_i)\rangle_{\mathcal{H}_m} + b,\ y_i\right) + \frac{1}{2}\sum_{m=1}^{M}\frac{||\boldsymbol{w}_m||^2_{\mathcal{H}_m}}{\theta_m} + \mu||\boldsymbol{\theta}||_p^2. \tag{18}$$

Setting the partial derivatives w.r.t. $\boldsymbol{\theta}$ to zero, we obtain the following condition on the optimality of $\boldsymbol{\theta}$,

$$-\frac{||\boldsymbol{w}_m||^2_{\mathcal{H}_m}}{2\theta_m^2} + \beta\cdot\frac{\partial\left(\frac{1}{2}||\boldsymbol{\theta}||_p^2\right)}{\partial\theta_m} = 0, \quad \forall m = 1,\ldots,M, \tag{19}$$

with non-zero β (it holds $\beta > 0$ by the strict convexity of $||\cdot||$). The first derivative of the ℓ_p-norm with respect to the mixing coefficients can be expressed as

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

and hence Eq. (19) translates into the following optimality condition,

$$\theta_m^* = \zeta||\boldsymbol{w}_m^*||^{\frac{2}{p+1}}_{\mathcal{H}_m}, \quad \forall m = 1,\ldots,M, \tag{20}$$

with a suitable constant ζ. By the strict convexity of $||\cdot||$ the constraint $||\boldsymbol{\theta}||_p^2 \leq 1$ in Optimization Problem (P) is at the upper bound and hence we have that $||\boldsymbol{\theta}^*||_p = 1$ for an optimal $\boldsymbol{\theta}^*$. Hence, ζ can be computed as $\zeta = \left(\sum_{m=1}^{M}||\boldsymbol{w}_m^*||^{2p/p+1}_{\mathcal{H}_m}\right)^{1/p}$. Combined with (20), this results in the claimed formula (17). ■

In the more interesting case, we will perform the above update in the dual, thereby operating on dual variables $\boldsymbol{\alpha}$:

Corollary 3 *Let V be a convex loss function, be $p > 1$. Given fixed dual variable $\boldsymbol{\alpha}$, as specified in Sect.* (2.3)*, the optimal solution of Optimization Problem (P) is attained for*

$$\theta_m^* = \frac{(\boldsymbol{\alpha}K_m\boldsymbol{\alpha})^{\frac{2}{p-1}}}{\left(\sum_{m'=1}^{M}(\boldsymbol{\alpha}K_{m'}\boldsymbol{\alpha})^{\frac{2p}{p-1}}\right)^{1/p}}, \qquad \forall m = 1,\ldots,M. \tag{21}$$

Note that if we deploy hinge loss, then we operate on variables $\alpha_i^{new} = \alpha_i y_i$ (cf. Sect. 3.1).

Proof According to Eq. (9b) the dual variables $\boldsymbol{\alpha}$ are specified in terms of $\boldsymbol{w}_m$ by

$$\boldsymbol{w}_m^* = \theta_m^*\sum_{i=1}^{n}\alpha_i^*\psi_m(\boldsymbol{x}_i).$$

Plugging the above primal-dual relations into Eq. (20) and appropriately normalizing, we obtain the desired dual update formula for $\boldsymbol{\theta}$. ■

Second we consider how to optimize Optimization Problem (P) w.r.t. the remaining variables $(\boldsymbol{w}_m, b)$ for a given, set of mixing coefficients $\boldsymbol{\theta}$. Since optimization often is considerably easier in the dual space, we fix $\boldsymbol{\theta}$ and build the partial Lagrangian of Optimization Problem (P) w.r.t. all other primal variables $\boldsymbol{w}$, b. The resulting dual problem is of the form

$$\sup_{\boldsymbol{\alpha}:\mathbf{1}^\top\boldsymbol{\alpha}=0} -C\sum_{i=1}^{n} V^*\left(-\frac{\alpha_i}{C},\ y_i\right) - \frac{1}{2}\sum_{m=1}^{M}\theta_m\boldsymbol{\alpha}^\top K_m\boldsymbol{\alpha}. \tag{22}$$

We now have all ingredients for an efficient ℓ_p-norm algorithm, based on alternatingly solving an SVM w.r.t. the actual mixture $\boldsymbol{\theta}$ and computing the analytical update according to Eq. (17). A simple wrapper algorithm is stated in Alg. 1.

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

1: **input** feasible $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$.
2: **while** optimality conditions are not satisfied **do**
3: solve Eq. (22), e.g., SVM, w.r.t. $\boldsymbol{\alpha}$
4: obtain updated $\boldsymbol{\theta}$ according to Eq. (21)
5: **end while**

A disadvantage of the above wrapper approach is that it deploys a full blown kernel matrix. Instead, we propose to interleave the SVM optimization of SVMlight with the $\boldsymbol{\theta}$- and $\boldsymbol{\alpha}$-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.[4] 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$^{\text{light}}$. Lines 6-8 compute (parts of) SVM-objective values for each kernel independently. Finally lines 10 to 14 solve the analytical $\boldsymbol{\theta}$-step. The algorithm terminates if the maximal KKT violation (c.f. Joachims, 1999) falls below a predetermined precision ε_{svm} and if the normalized maximal constraint violation $|1 - \frac{S^t}{\lambda}| < \varepsilon_{mkl}$ for the MKL-step.

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

Proposition 4 (Bertsekas, 1999) *Let $\mathcal{X} = \bigotimes_{m=1}^{M}\mathcal{X}_m$ be a the Cartesian product of closed convex sets $\mathcal{X}_m \in \mathbb{R}^{d_m}$, be $f : \mathcal{X} \to \mathbb{R}$ a continuously differentiable function. Define the non-linear Gauss-Seidel method recursively by letting $x^0 \in \mathcal{X}$ be any feasible point, and be*

$$x_m^{k+1} = \operatorname*{argmin}_{\xi\in\mathcal{X}_m} f\left(x_1^{k+1}, \cdots, x_{m-1}^{k+1}, \xi, x_{m+1}^{k}, \cdots, x_M^{k}\right), \quad \forall m = 1, \ldots, M. \tag{23}$$

4. In practice, it turns out that the kernel matrix of active variables usually 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 $\boldsymbol{\theta}$ and SVM α are optimized interleavingly. The accuracy parameter ϵ and the subproblem size Q are assumed to be given to the algorithm.*

$g_{m,i} = 0$, $\hat{g}_i = 0$, $\alpha_i = 0$, $\theta_m = \sqrt[p]{1/M}$ for $m = 1, \ldots, M$ and $i = 1, \ldots, n$
for $t = 1, 2, \ldots$ and while SVM and MKL optimality conditions are not satisfied **do**
  Select Q suboptimal variables $\alpha_{i_1}, \ldots, \alpha_{i_Q}$ based on the gradient $\hat{\mathbf{g}}$ and $\boldsymbol{\alpha}$; store $\boldsymbol{\alpha}^{old} = \boldsymbol{\alpha}$
  Solve SVM dual with respect to the selected variables and update $\boldsymbol{\alpha}$
  Update gradient $g_{m,i} \leftarrow g_{m,i} + \sum_{q=1}^{Q} (\alpha_{i_q} - \alpha_{i_q}^{old}) y_{i_q} k_m(\boldsymbol{x}_{i_q}, \boldsymbol{x}_i)$ for all $m = 1, \ldots, M$ and $i = 1, \ldots, n$
  for $m = 1, \ldots, M$ **do**
    $S_m^t = \frac{1}{2} \sum_i g_{m,i} \alpha_i y_i$
  end for
  if $|1 - \frac{S^t}{\lambda}| \geq \epsilon$
    while MKL optimality conditions are not satisfied **do**
      for $m = 1, \ldots, M$
        $\boldsymbol{\theta}_m = (S_m^t)^{1/(p+1)} \,/\, \left(\sum_{m'=1}^{M} (S_{m'}^t)^{p/(p+1)} \right)^{1/p}$
      end for
    end while
  end if
  $\hat{g}_i = \sum_m \theta_m g_{m,i}$ for all $i = 1, \ldots, n$
end for

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

$$\min_{\xi \in \mathcal{X}_m} f\left(x_1, \cdots, x_{m-1}, \xi, x_{m+1}, \cdots, x_M\right) \tag{24}$$

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 the proposed ℓ_p-norm MKL training algorithm.

Proposition 5 *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 (P). 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 coincidence for the hinge loss. Hence, it suffices to show the wrapper algorithm converges.

To this aim, we have to transform Optimization Problem (P) into a form such that the requirements for application of Prop. 4 are fulfilled. We start by expanding Optimization Problem (P) into

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

thereby extending the second block of variables, $(\boldsymbol{w}, b)$, into $(\boldsymbol{w}, b, \boldsymbol{\xi})$. Moreover, we note that after an application of the representer theorem[5] (Kimeldorf and Wahba, 1971) we may without loss of generality assume $\mathcal{H}_m = \mathbb{R}^n$.

In above problem's current form, the possibility of $\theta_m = 0$ while $\boldsymbol{w}_m \neq 0$ renders the objective function nondifferentiable, and it can take on infinite values. This hinders the application of the Prop. 4. Fortunately, in the optimal point, we always have $\theta_m* > 0$ for all m, which can be verified by Eq. (21), where we use the positive definiteness of the kernel matrices K_m. We therefore can substitute the constraint $\boldsymbol{\theta} \geq 1$ by $\boldsymbol{\theta} > 1$ for all m. In order to maintain the closeness of the feasible set we subsequently apply a bijective coordinate transformation $\phi_m : \mathbb{R}_+^M \to \mathbb{R}^M$ with $\phi(\theta_m) = \log(\theta_m)$, resulting in the following equivalent problem,

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

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

Applying the Gauss-Seidel method in Eq. (23) to the base problem (P) and to the reparametrized problem yields the same sequence of solutions $\{(\boldsymbol{w}, b, \boldsymbol{\theta})^k\}_{k\in\mathbb{N}_0}$. Fortunately, the above problem now allows to apply Prop. 4 for the two blocks of coordinates $\boldsymbol{\theta} \in \mathcal{X}_1$ and $(\boldsymbol{w}, b, \boldsymbol{\xi}) \in \mathcal{X}_2$: the objective is continuously differentiable and the sets $\mathcal{X}_1$ are closed and convex. To see the latter, note that $||\cdot||_p^2 \circ \exp$ is a convex function since $||\cdot||_p^2$ is convex and non-increasing in each argument (cf., e.g., Section 3.2.4 of Boyd and Vandenberghe, 2004). Moreover, the minima in Eq. (23) are uniquely attained: the $(\boldsymbol{w}, b)$-step amounts to solving an SVM on a positive definite kernel mixture, and the analytical $\boldsymbol{\theta}$-step clearly yields unique solutions as well.

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

In practice, we are facing two problems. Firstly, the standard Hilbert space setup necessarily implies that $\|\boldsymbol{w}_m\| \geq 0$ 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 $\|\boldsymbol{w}_m\| \leq 0$ it also follows that $\theta_m^\star = 0$ as long as at least one strictly positive $\|\boldsymbol{w}_{m'}\| > 0$ 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 $\|\boldsymbol{w}_m\| \leq 0$, we can immediately set the corresponding mixing coefficients $\theta_m^\star$ to zero. The remaining $\boldsymbol{\theta}$ are then computed according to Equation (2), and convergence will be achieved as long as at least one strictly positive $\|\boldsymbol{w}_{m'}\| > 0$ exists in each iteration.

Secondly, 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 $\boldsymbol{\alpha}$ only a

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

little bit before recomputing $\boldsymbol{\theta}$ since computing a high precision solution can be wasteful, as indicated by the superior performance of the interleaved algorithms (cf. Sect. 5.5). This helps to avoid spending a lot of $\boldsymbol{\alpha}$-optimization (SVM training) on a suboptimal mixture $\boldsymbol{\theta}$. Fortunately, we can overcome the potential convergence problem by ensuring that the primal objective decreases within each $\boldsymbol{\alpha}$-step. Then the alternating optimization is guaranteed to converge. 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 5).

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.

4.2 Cutting Planes

In order to obtain an alternative optimization strategy, we fix $\boldsymbol{\theta}$ in the primal MKL optimization problem (P) and build the partial Lagrangian w.r.t. all other primal variables $\boldsymbol{w}$, b. The resulting dual problem is a min-max problem of the form

$$\inf_{\boldsymbol{\theta}:\ \boldsymbol{\theta}\geq\mathbf{0},\ ||\boldsymbol{\theta}||^2\leq 1} \quad \sup_{\boldsymbol{\alpha}:\mathbf{1}^\top\boldsymbol{\alpha}=0} \quad -C\sum_{i=1}^{n} V^*\left(-\frac{\alpha_i}{C},\ y_i\right) - \frac{1}{2}\sum_{m=1}^{M}\theta_m\boldsymbol{\alpha}^\top K_m\boldsymbol{\alpha} \tag{25}$$

We focus on the hinge loss, i.e., $V(t,y) = \max(0, 1-ty)$, and non-sparse norms of the form $||\boldsymbol{\theta}|| = \left(\sum_{m=1}^{M}\theta_m^p\right)^{1/p}$ (nevertheless, the following reasoning holds for every twice differentiable norm). Thus, employing a variable substition of the form $\alpha_i^{\text{new}} = \alpha_i y_i$, Eq. (25) translates into

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

where $Q_j = YK_jY$ for $1\leq j\leq m$ and $Y=\operatorname{diag}(\mathbf{y})$. The above optimization problem is a *saddle point problem* and can be solved by alternating $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ optimization step. While the former can simply be carried out by a support vector machine for a fixed mixture $\boldsymbol{\theta}$, the latter has been optimized for $p=1$ by reduced gradients (Rakotomamonjy et al., 2007).

We take a different approach and translate the min-max problem into an equivalent semi-infinite program (SIP) as follows. Denote the value of the target function by $t(\boldsymbol{\alpha},\boldsymbol{\theta})$ and suppose $\boldsymbol{\alpha}^*$ is optimal. Then, according to the max-min inequality (Boyd and Vandenberghe, 2004, p. 115), we have $t(\boldsymbol{\alpha}^*,\boldsymbol{\theta})\geq t(\boldsymbol{\alpha},\boldsymbol{\theta})$ for all $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$. Hence, we can equivalently minimize an upper bound η on the optimal value and arrive at the following semi-infinite

Algorithm 3 *Chunking-based ℓ_p-Norm MKL cutting plane training algorithm. It simultaneously optimizes the variables $\boldsymbol{\alpha}$ and the kernel weighting $\boldsymbol{\theta}$. The accuracy parameter ϵ and the subproblem size Q are assumed to be given to the algorithm. For simplicity, a few speed-up tricks are not shown, e.g., hot-starts of the SVM and the QCQP solver.*

$g_{m,i} = 0$, $\hat{g}_i = 0$, $\alpha_i = 0$, $\theta_m = \sqrt[p]{1/M}$ for $m = 1, \ldots, M$ and $i = 1, \ldots, n$
for $t = 1, 2, \ldots$ and while SVM and MKL optimality conditions are not satisfied **do**
  Select Q suboptimal variables $\alpha_{i_1}, \ldots, \alpha_{i_Q}$ based on the gradient $\hat{\mathbf{g}}$ and $\boldsymbol{\alpha}$; store $\boldsymbol{\alpha}^{old} = \boldsymbol{\alpha}$
  Solve SVM dual with respect to the selected variables and update $\boldsymbol{\alpha}$
  Update gradient $g_{m,i} \leftarrow g_{m,i} + \sum_{q=1}^{Q} (\alpha_{i_q} - \alpha_{i_q}^{old}) y_{i_q} k_m(\boldsymbol{x}_{i_q}, \boldsymbol{x}_i)$ for all $m = 1, \ldots, M$ and $i = 1, \ldots, n$
  for $m = 1, \ldots, M$ **do**
    $S_m^t = \frac{1}{2} \sum_i g_{m,i} \alpha_i y_i$
  end for
  $L^t = \sum_i \alpha_i, \quad S^t = \sum_m \theta_m S_m^t$
  if $|1 - \frac{S^t}{\lambda}| \geq \epsilon$
    while MKL optimality conditions are not satisfied **do**
      $\boldsymbol{\theta}^{old} = \boldsymbol{\theta}$
      $(\boldsymbol{\theta}, \lambda) \leftarrow$ argmax λ
        w.r.t. $\boldsymbol{\theta} \in \mathbb{R}^M, \lambda \in \mathbb{R}$
        s.t. $\mathbf{0} \leq \boldsymbol{\theta} \leq \mathbf{1}$,
          $\frac{p(p-1)}{2} \sum_m (\theta_m^{old})^{p-2} \theta_m^2 - \sum_m p(p-2)(\theta_m^{old})^{p-1} \theta_m \leq \frac{p(3-p)}{2}$ and
          $\sum_m \theta_m S_m^r - L^r \geq \lambda$ for $r = 1, \ldots, t$
      $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} / ||\boldsymbol{\theta}||_p$
      Remove inactive constraints
    end while
  end if
  $\hat{g}_i = \sum_m \theta_m g_{m,i}$ for all $i = 1, \ldots, n$
end for

program,

$$\begin{aligned} \min_{\eta} \quad & \eta \\ \text{s.t.} \quad & \forall \boldsymbol{\alpha} \in \mathcal{A}: \quad \eta \geq \mathbf{1}^\top \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^\top \sum_{m=1}^{M} \theta_m Q_m \boldsymbol{\alpha}; \\ & \boldsymbol{\theta} \geq \mathbf{0}; \quad \|\boldsymbol{\theta}\|_p^2 \leq 1, \end{aligned} \tag{SIP}$$

where $\mathcal{A} = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^n \mid \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}, \boldsymbol{y}^\top \boldsymbol{\alpha} = 0 \right\}$.

Sonnenburg et al. (2006a) optimize the above SIP for $p = 1$ with interleaving cutting plane algorithms. The solution of a quadratic program (here the regular SVM) generates the most strongly violated constraint for the actual mixture $\boldsymbol{\theta}$. The optimal $(\boldsymbol{\theta}^*, \eta)$ is then identified by solving a linear program with respect to the set of active constraints. The optimal mixture is then used for computing a new constraint and so on.

Unfortunately, for $p > 1$, a non-linearity is introduced by requiring $\|\boldsymbol{\theta}\|_p^2 \leq 1$ and such constraint is unlikely to be found in standard optimization toolboxes that often handle only linear and quadratic constraints. As a remedy, we propose to approximate the constraint

$\|\boldsymbol{\theta}\|_p^p \leq 1$ by a sequence of second-order Taylor expansions[6]

$$\begin{aligned}||\boldsymbol{\theta}||_p^p &\approx ||\tilde{\boldsymbol{\theta}}||_p^p + p\left(\tilde{\boldsymbol{\theta}}^{p-1}\right)^\top \left(\boldsymbol{\theta}-\tilde{\boldsymbol{\theta}}\right) + \frac{p(p-1)}{2}\left(\boldsymbol{\theta}-\tilde{\boldsymbol{\theta}}\right)^\top \operatorname{diag}\left(\tilde{\boldsymbol{\theta}}^{p-2}\right)\left(\boldsymbol{\theta}-\tilde{\boldsymbol{\theta}}\right)\\ &= 1 + \frac{p(p-3)}{2} - \sum_{m=1}^{M} p(p-2)(\tilde{\theta}_m)^{p-1}\,\theta_m \;+ \frac{p(p-1)}{2}\sum_{m=1}^{M}\tilde{\theta}_m^{p-2}\,\theta_m^2,\end{aligned}$$

where $\boldsymbol{\theta}^p$ is defined element-wise, that is $\boldsymbol{\theta}^p := (\theta_1^p, \ldots, \theta_M^p)$. The sequence $(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \cdots)$ is initialized with a uniform mixture satisfying $||\boldsymbol{\theta}_0||_p^p = 1$ as a starting point. Successively $\boldsymbol{\theta}_{t+1}$ is computed using $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}_t$. Note that the Hessian of the quadratic term in the approximation is diagonal, strictly positiv and very-well conditioned wherefore the resulting quadratically constrained problem can be solved efficiently. In fact, since there is only one quadratic constraint, its complexity should rather be compared to that of a considerably easier quadratic program. Moreover, in order to ensure convergence, we enhanced the resulting sequential quadratically constrained quadratic programming by projection steps onto the boundary of the feasible set, as given in Line 19. Finally note, that this approach can be further sped-up by additional level-set projections in the $\boldsymbol{\theta}$-optimization phase similar to Xu et al. (2009). In our case, the level-set projection is a convex quadratic problem with ℓ_p-norm constraints and can again be approximated by a successive sequence of second-order Taylor expansions.

Algorithm 3 outlines the interleaved $\boldsymbol{\alpha}, \boldsymbol{\theta}$ MKL training algorithm. Lines 3-5 are standard in chunking based SVM solvers and carried out by SVM$^{\text{light}}$. Lines 6-9 compute (parts of) SVM-objective values for each kernel independently. Finally lines 11 to 19 solve a sequence of semi-infinite programs with the ℓ_p-norm constraint being approximated as a sequence of second-order constraints. The algorithm terminates if the maximal KKT violation (see Joachims, 1999) falls below a predetermined precision ε_{svm} and if the normalized maximal constraint violation $|1 - \frac{S^t}{\lambda}| < \varepsilon_{mkl}$ for the MKL. The following proposition shows the convergence of the semi-infinite programming loop in Algorithm 3.

Proposition 6 *Let the kernel matrices $K_1, \ldots, K_M$ be positive definite and be $p > 1$. Suppose that the SVM computation is solved exactly in each iteration. Moreover, suppose there exists an optimal limit point of nested sequence of QCCPs. Then the sequence generated by Algorithm 3 has at least one point of accumulation that solves Optimization Problem (P).*

Proof By assumption the SVM is solved to infinite precision in each MKL step which simplifies our analysis in that the numerical details in Algorithm 3 can be ignored. We conclude, that the outer loop of Alg. 3 amounts to a cutting-plane algorithm for solving the semi-infinite program (SIP). It is well-known (Sonnenburg et al., 2006a), that this algorithm converges, in the sense that there exists at least one point of accumulation, which solves the primal problem (P). E.g. this can be seen by viewing the cutting plane algorithm as a special instance of the class of so-called *exchange methods* and subsequently applying Theorem 7.2 in Hettich and Kortanek (1993). A difference to the analysis in Sonnenburg et al. (2006a) is the $\ell_{p>1}$-norm constraint in our algorithm. However, according to our assumption that the nonlinear subprogram is solved correctly, a quick inspection

6. We also tried out first-order Taylor expansions, whereby our algorithm basically boils down the renowned *sequential quadratic programming*, but it empirically turned out to be inferior. Intuitively, second-order expansions work best when the approximated function is almost quadratic, as given in our case.

of the preliminaries of the latter theorem clearly reveals, that they remain fulfilled when introducing an ℓ_p-norm constraint. ■

In order to complete our convergence analysis, it remains to show that the inner loop (lines 11-18), that is the sequence of QCQPs, converges against an optimal point. Existing analyses of this so-called *sequential quadratically constrained quadratic programming* (SQCQP) can be divided into two classes. First, one class establishes *local* convergence, i.e., convergence in an open neighborhood of the optimal point, at a rate of $O(n^2)$, under relatively mild smoothness and constraint qualification assumptions (Fernández and Solodov, 2008; Anitescu, 2002), whereas Anitescu (2002) additionally requires quadratic growth of the nonlinear constraints. Those analyses basically guarantee local convergence the nested sequences of QCQPs in our ℓ_p-norm training algorithm, for all $p \in (1, \infty)$ (Fernández and Solodov, 2008) and $p \geq 2$ (Anitescu, 2002), respectively.

A second class of papers additionally establishes *global* convergence (e.g. Solodov, 2004; Fukushima et al., 2002), so they need more restrictive assumptions. Moreover, in order to ensure feasibility of the subproblems when the actual iterate is too far away from the true solution, a modification of the algorithmic protocol is needed. This is usually dealt by performing a subsequent line search and downweighting the quadratic term by a multiplicative adaptive constant $D_i \in [0, 1]$. Unfortunately, the latter involves a complicated procedure to tune D_i (Fukushima et al., 2002, p. 7). Employing the above modifications, the analysis in Fukushima et al. (2002) together with our Prop. 6 would guarantee the convergence of our Alg. 3.

However, due to the special form of our SQCQP, we chose to discard the comfortable convergence guarantees and to proceed with a much more simple and efficient strategy, which renders both the expensive line search and the tuning of the constant D_i unnecessary. The idea of our method is that the projection of $\boldsymbol{\theta}$ onto the boundary of the feasible set, given by line 18 in Alg. 3, can be performed analytically. This projection ensures the feasibility of the QCQP subproblems. Note that in general, this projection can be as expensive as performing a QCQP step, which is why projection-type algorithms for solving SQCQPs to the best of our knowledge have not been studied yet by the optimization literature.

Although the projection procedure is appealingly simple and—as we found empirically—seemingly shares nice convergence properties (the sequence of SQCQPs converged optimally in all cases we tried, usually after 3-4 iterations), it unfortunately prohibits exploitation of existing analyses for global convergence. However, the discussions in Fukushima et al. (2002) and Solodov (2004) identify the reason of occasional divergence of the vanilla SQCQP as the infeasibility of the subproblems. But in contrast, our projection algorithm always ensures the feasibility of the subproblem. We therefore conjecture that based on the superior empirical results and the discussions in Fukushima et al. (2002) and Solodov (2004), our algorithm is designated to convergence. The theoretical analysis of this new class of so-called *SQCQP projection algorithms* is beyond the scope of this paper.

4.3 Technical Considerations

4.3.1 Implementation Details

We have implemented the analytic and the cutting plane algorithm as well as a Newton method (c.f. Kloft et al., 2009a) within the SHOGUN toolbox[7] for regression, one-class classification, and two-class classification tasks. In addition one can choose the optimization scheme, i.e., 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.

In the more conventional family of approaches, the so-called *wrapper algorithms*, an optimization schme on $\boldsymbol{\theta}$ wraps around a single kernel SVM. Effectively this results in alternatingly solving for $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$. For the outer optimization (i.e., that on $\boldsymbol{\theta}$) 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[8] or IBM ILOG CPLEX[9]. 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(`$\sum_\alpha$`, obj`$_m$`)`, that has the arguments $\sum_{\boldsymbol{\alpha}} = \sum_{i=1}^n \alpha_i$ and $\text{obj}_m = \frac{1}{2}\boldsymbol{\alpha}^T K_m \boldsymbol{\alpha}$, i.e. the current linear $\boldsymbol{\alpha}$-term and the SVM objectives for for each kernel. This function is either, in the 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 $\sum_{\boldsymbol{\alpha}} = 0$ 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 $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$. Thus one needs direct access to $\boldsymbol{\alpha}^*$ and computes $\sum_{\boldsymbol{\alpha}} = -\sum_{i=1}^n (\alpha_i + \alpha_i^*)\varepsilon - \sum_{i=1}^n (\alpha_i - \alpha_i^*) y_i$ (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.[10] While for the wrapper algorithm 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.

7. `http://www.shogun-toolbox.org`.
8. `http://www.gnu.org/software/glpk/`.
9. `http://www.ibm.com/software/integration/optimization/cplex/`.
10. *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.

4.3.2 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 that, moreover, 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 carries the rescaling of *data points* to the world of kernels. 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 λ_m of the kernel, such that the rescaled kernel $\tilde{k}_m(\cdot,\cdot) = \lambda_m k_m(\cdot,\cdot)$ and the corresponding feature map $\tilde{\Phi}_m(\cdot) = \sqrt{\lambda_m}\Phi_m(\cdot)$ satisfy

$$1 \quad \stackrel{!}{=} \quad \frac{1}{n}\sum_{i=1}^{n}\left(\tilde{\Phi}_m(\boldsymbol{x}_i) - \tilde{\Phi}_m(\bar{\boldsymbol{x}})\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\tilde{k}_m(\boldsymbol{x}_i,\boldsymbol{x}_i) - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\tilde{k}_m(\boldsymbol{x}_i,\boldsymbol{x}_j)$$

for each $m = 1,\ldots,M$, where $\tilde{\Phi}_m(\bar{\boldsymbol{x}}) := \frac{1}{n}\sum_{i=1}^{n}\tilde{\Phi}_m(\boldsymbol{x}_i)$ is the empirical mean of the data in feature space. The final normalization rule is

$$k(\boldsymbol{x},\bar{\boldsymbol{x}}) \longmapsto \frac{k(\boldsymbol{x},\bar{\boldsymbol{x}})}{\frac{1}{n}\sum_{i=1}^{n}k(\boldsymbol{x}_i,\boldsymbol{x}_i) - \frac{1}{n^2}\sum_{i,j=1}^{n},k(\boldsymbol{x}_i\bar{\boldsymbol{x}}_j)}. \tag{26}$$

Spherical Normalization. Frequently, kernels are normalized according to

$$k(\boldsymbol{x},\bar{\boldsymbol{x}}) \longmapsto \frac{k(\boldsymbol{x},\bar{\boldsymbol{x}})}{\sqrt{k(\boldsymbol{x},\boldsymbol{x})k(\bar{\boldsymbol{x}},\bar{\boldsymbol{x}})}}. \tag{27}$$

After this operation, $\|\boldsymbol{x}\| = k(\boldsymbol{x},\boldsymbol{x}) = 1$ 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: in case the kernel is centered (i.e. average of the data points lies on the origin), the rescaled kernel satisfies the above goal that the points have unit variance (around their mean). Thus the spherical normalization may be seen as an approximation 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.

4.4 Relation to Block-Norm Formulation and Limitations of Our Framework

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

$$\inf_{\boldsymbol{w},b,\boldsymbol{\theta}:\boldsymbol{\theta}\geq\mathbf{0}} \quad C\sum_{i=1}^{n} V\left(\sum_{m=1}^{M}\langle \boldsymbol{w}_m, \psi_m(\boldsymbol{x}_i)\rangle_{\mathcal{H}_m} + b,\ y_i\right) + \frac{1}{2}\sum_{m=1}^{M}\frac{||\boldsymbol{w}_m||^2_{\mathcal{H}_m}}{\theta_m}, \quad \text{s.t.} \quad ||\boldsymbol{\theta}||_p^2 \leq 1. \tag{28}$$

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

$$\inf_{\boldsymbol{w},b,\boldsymbol{\theta}:\boldsymbol{\theta}\geq\mathbf{0}} \quad \tilde{C}\sum_{i=1}^{n} V\left(\sum_{m=1}^{M}\langle \boldsymbol{w}_m, \psi_m(\boldsymbol{x}_i)\rangle_{\mathcal{H}_m} + b,\ y_i\right) + \frac{1}{2}\sum_{m=1}^{M}||\boldsymbol{w}_m||^q_{\mathcal{H}_m}, \tag{29}$$

where $q = \frac{2p}{p+1}$, and $\tilde{C}$ is a constant. For $q = 1$ this has been studied by Bach et al. (2004).

Proposition 7 *Let be $p > 1$ and be V a convex loss function. Optimization Problem* (28) *and* (29) *are equivalent, i.e., for each C there exists a $\tilde{C} > 0$, such that for each optimal solution $(\boldsymbol{w}^*, b^*, \theta^*)$ of OP* (28) *using C, we have that $(\boldsymbol{w}^*, b^*)$ is also optimal in OP* (29) *using $\tilde{C}$, and vice versa.*

Proof We begin by applying Theorem 1 to rephrase Optimization Problem (P) as

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

Setting the partial derivatives w.r.t. $\boldsymbol{\theta}$ to zero, we obtain the following equation at optimality:

$$-\frac{||\boldsymbol{w}_m||^2_{\mathcal{H}_m}}{2\theta_m^2} + \beta\cdot\theta_m^{p-1}||\boldsymbol{\theta}||_p^{2-p} = 0, \quad \forall m = 1,\ldots,M. \tag{30}$$

Hence, Eq. (30) translates into the following optimality condition on $\boldsymbol{w}$ and $\boldsymbol{\theta}$:

$$\theta_m^* = \zeta||\boldsymbol{w}_m^*||^{\frac{2}{p+1}}_{\mathcal{H}_m}, \quad \forall m = 1,\ldots,M,$$

with a suitable constant ζ. Plugging the above equation into Optimization Problem (P) yields

$$\inf_{\boldsymbol{w},b,\boldsymbol{\theta}:\boldsymbol{\theta}\geq\mathbf{0}} \quad C\sum_{i=1}^{n} V\left(\sum_{m=1}^{M}\langle \boldsymbol{w}_m, \psi_m(\boldsymbol{x}_i)\rangle_{\mathcal{H}_m} + b,\ y_i\right) + \frac{1}{2\zeta}\sum_{m=1}^{M}||\boldsymbol{w}_m||^{\frac{2p}{p+1}}_{\mathcal{H}_m}. \tag{31}$$

Defining $q := \frac{2p}{p+1}$ and $\tilde{C} := \zeta C$ results in (29) what was to show. ■

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 $[1,2]$. This raises the

question whether we can derive an efficient wrapper-based optimization strategy for the case of $q > 2$. A framework by Aflalo et al. (2010) covers the case $q \geq 2$, although their method aims at hierarchical kernel learning. Note, that $q \leq 2$ and hence ℓ_p-norm MKL is not covered by their approach.

We briefly sketch the analysis of Aflalo et al. (2010) and discuss a potential simplification of their algorithm for the special case of $\ell_{q>2}$ block norm MKL. We start by noting that it is possible to show that for $q \geq 2$, Eq. (29) is equivalent to

$$\sup_{\boldsymbol{\theta}:\boldsymbol{\theta}\geq\mathbf{0},||\boldsymbol{\theta}||_p^2\leq 1} \inf_{\boldsymbol{w},b} \quad \tilde{C}\sum_{i=1}^{n} V\left(\sum_{m=1}^{M}\langle \boldsymbol{w}_m, \psi_m(\boldsymbol{x}_i)\rangle_{\mathcal{H}_m} + b,\ y_i\right) + \frac{1}{2}\sum_{m=1}^{M}\theta_m||\boldsymbol{w}_m||_{\mathcal{H}_m}^2, \qquad (32)$$

where $p = \frac{q/2}{q/2-1}$. Note that despite the similarity to ℓ_p-norm MKL, the above problem significantly differs from ℓ_p-norm MKL for two reasons. Firstly, obvious differences such as the mixing coefficients $\boldsymbol{\theta}$ appearing in the nominator and the consequential maximization w.r.t. $\boldsymbol{\theta}$, render the above problem a min-max problem. Secondly, note that by varying p in the interval $[1, \infty]$, the whole range of q in the interval $[2, \infty]$ can be obtained, which explains why this method is complementary to ours, where q ranges in $[1, 2]$.

Using the hinge loss, Eq. (32) can be partially dualized w.r.t. fixed $\boldsymbol{\theta}$, resulting in a convex optimization problem (Boyd and Vandenberghe, 2004, p. 76)

$$\begin{aligned} \max_{\boldsymbol{\alpha},\boldsymbol{\theta}} \quad & \mathbf{1}^\top\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^\top\sum_{m=1}^{M}\frac{Q_m}{\theta_m}\boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}; \quad \boldsymbol{y}^\top\boldsymbol{\alpha} = 0; \quad \boldsymbol{\theta} \geq 0; \quad \|\boldsymbol{\theta}\|_p^2 \leq 1, \end{aligned} \qquad (33)$$

where, as in the previous sections, we denote $Q_j = YK_jY$ and and $Y = \text{diag}(\mathbf{y})$. Originally the authors aimed at hierarchical kernel learning and Aflalo et al. (2010) proposed to optimize (33) by a mirror descent algorithm (Beck and Teboulle, 2003). However, for the special case of $q > 2$ block norm MKL, which we consider here, a simple block gradient procedure based on an analytical update of $\boldsymbol{\theta}$, similar to the one presented in Section 4.1, is sufficient. We omit the derivations which are analogeous to those presented in Section 4.1.

5. Computational Experiments

In this section we study non-sparse MKL in terms of computational efficiency and predictive accuracy. Throughout all our experiments both ℓ_p-norm MKL implementations, presented in Sections 4.1 and 4.2, perform comparably. We apply the method of (Sonnenburg et al., 2006a) in the case of $p = 1$, as it is recovered as a special case of our cutting plane strategy. We write ℓ_∞-norm MKL for a regular SVM with the unweighted-sum kernel $K = \sum_m K_m$.

We first study a toy problem in Section 5.1 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 the Bioninformatics domain, namely protein subcellular localization (Section 5.2), finding transcription start sites of RNA Polymerase II binding genes in genomic DNA sequences (Section 5.3), and reconstructing metabolic gene networks (Section 5.4).

Complementarily, we would like to mention empirical results of other researchers which have been experimenting with non-sparse MKL. Cortes et al. (2009) applies ℓ_2-norm MKL to regression tasks on Reuters and various sentiment analysis datasets, and Yu et al. (2009) studies ℓ_2-norm on two real-world genomic data sets for clinical decision support in cancer diagnosis and disease relevant gene prioritization, respectively. Yan et al. (2009) apply ℓ_2-norm MKL to image and video classification tasks. All those papers show an improvement of ℓ_2-norm MKL over sparse MKL and the unweighted sum kernel SVM. Nakajima et al. (2009) study ℓ_p-norm MKL for multi-label image categorization and show an improvement of non-sparse MKL over $\ell_{1/\infty}$-norm MKL.

5.1 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 learnt to the chosen norm p in the model. It is suggestive 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 focussed 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 $[0, \infty]$.

We generate an n-elemental balanced sample $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ from two $d = 50$-dimensional isotropic Gaussian distributions with equal covariance matrices $C = I_{d\times d}$. The two Gaussians are aligned at opposing means w.r.t. to the origin, $\mu_1 = \frac{\rho}{||\boldsymbol{\theta}||_2}\boldsymbol{\theta}$ and $\mu_2 = -\frac{\rho}{||\boldsymbol{\theta}||_2}\boldsymbol{\theta}$. Thereby $\boldsymbol{\theta}$ is a binary vector, i.e., $\theta_i \in \{0, 1\}$, encoding the true underlying data sparsity as follows. Zero components $\theta_i = 0$ clearly imply identical means of the two classes distributions in the i-th feature set; hence the latter does not carry any discriminating information. In summary, the fraction of zero components, $\nu(\boldsymbol{\theta}) = 1 - \frac{1}{d}\sum_{i=1}^d \boldsymbol{\theta}_i$, is a measure for the feature sparsity of the learning problem.

For several values of ν we generate $m = 250$ data sets $\mathcal{D}_1, \ldots, \mathcal{D}_m$ 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 4.3.2. Hence, the $\nu(\boldsymbol{\theta})$ 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 by grid search over $C \in 10^{[-4,3.5,\ldots,0]}$ (optimal Cs are attained in the interior of the grid). We report on test errors evaluated on $10,000$-elemental independent test sets and pure mean model errors of the computed kernel mixtures, that is $\Delta\boldsymbol{\theta} = ||\zeta(\hat{\boldsymbol{\theta}}_{\mathrm{MKL}}) - \zeta(\boldsymbol{\theta})||_2$, where $\zeta(\boldsymbol{x}) = \frac{\boldsymbol{x}}{||\boldsymbol{x}||_2}$.

The results are shown in Fig. 1 for $n = 50$ and $n = 800$ where the figures on the left show the test error and the ones on the right the model error $\Delta\boldsymbol{\theta}$. Regarding the latter, model errors reflecting 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.

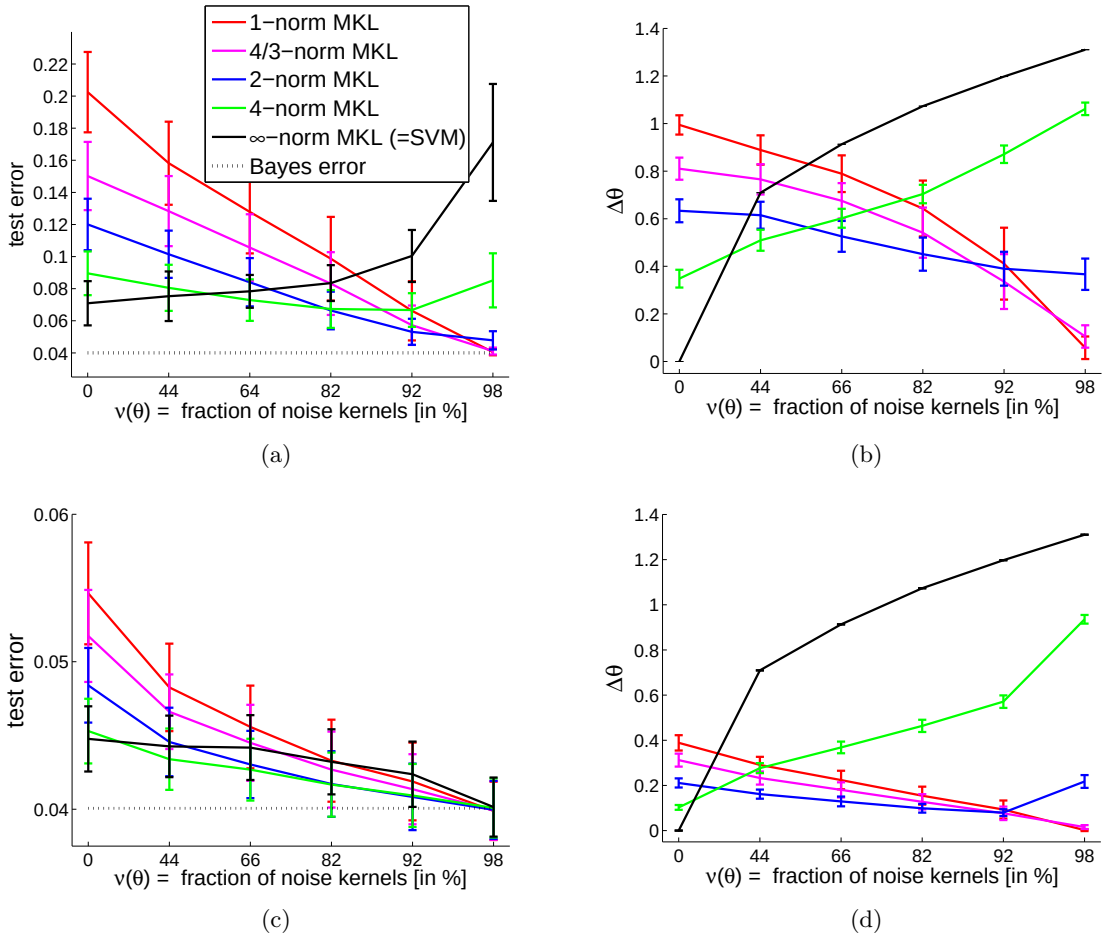

Figure 1: Results of the artificial experiment for sample sizes of $n = 50$ (top) and $n = 800$ (below) training instances in terms of test errors (left) and mean model errors $\Delta\boldsymbol{\theta}$ (right).

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. In contrast, the vanilla SVM on an unweighted sum kernel performs best when all kernels are equally informative, however its performance does not approach the Bayes error rate for the reasons discussed in Sect. 4.4. The non-sparse ℓ_4- and ℓ_2-norm MKL variants perform best in the balanced scenarios, i.e., 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 an test error of less than 0.1% in all scenarios. The sparse ℓ_1-norm MKL performs worst when the noise level is less than 82%. 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 MKL variants perform well in the sparse scenarios. In

contrast to the test errors, the mean model errors depicted in Figure 1 (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. 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 robustly.

In summary, the choice of the norm parameter p is important for small sample sizes while 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.

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

We downloaded the kernel matrices of all 4 data sets[11]. The kernel matrices are multiplicatively normalized as described in Section 4.3.2. The experimental setup used here is related to that of Ong and Zien (2008), although it deviates from it in several details. 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 a MKL-SVM 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. Instead we determine for each p the value of C that yields the highest MCC. Finally, we obtain an optimized C and MCC value for each combination of data set, split, and norm p.

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. This is remarkable, as this data set is particular in that it fulfills the rare condition that ℓ_1-norm MKL performs better than ℓ_∞-norm MKL. In other words, selecting these data may imply a bias towards ℓ_1-norm. 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.

11. Available from `http://www.fml.tuebingen.mpg.de/raetsch/suppl/protsubloc/`

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, where the average is taken over the 30 splits of the data, and for each split the MCC is maximized w.r.t. C (i.e., C is selected to be optimal).

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.	±0.47	±0.45	±0.43	±0.42	±0.42	±0.43	±0.47	±0.49	±0.55	±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.	±0.26	±0.25	±0.26	±0.27	±0.29	±0.32	±0.32	±0.32	±0.32	±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.	±0.35	±0.34	±0.34	±0.34	±0.33	±0.32	±0.32	±0.34	±0.33	±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.	±0.66	±0.63	±0.67	±0.62	±0.61	±0.67	±0.72	±0.81	±0.83	±0.80

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 analoguous to considering (inhomogeneous) polynominal 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 the exclusive parts of information are not very discriminative. Hence each set of 16 kernels is highly redundant. This renders a non-sparse kernel mixture ineffective. We conclude that ℓ_1-norm must be the best prediction model.

5.3 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 utilizing dbTSS version 4 (Suzuki et al., 2002) and refseq genes. These are translated into

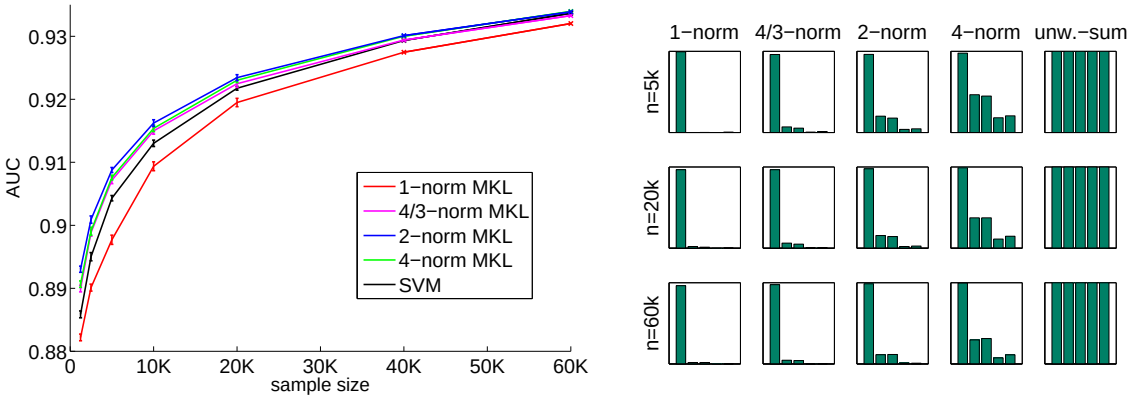

Figure 2: (left) Area under ROC curve (AUC) on test data for TSS recognition as a function of the training set size. Notice the tiny bars indicating standard errors w.r.t. repetitions on disjoint training sets. (right) Corresponding kernel mixtures. For $p = 1$ consistent sparse solutions are obtained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

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 4.3.2. We reserve 13,000 and 20,000 randomly drawn instances for holdout and test sets, respectively, and use the remaining 60,000 as the training pool. Figure 2 shows test errors for varying training set sizes drawn from the pool; training sets of the same size are disjoint. Error bars indicate standard errors of repetitions for small training set sizes.

Regardless of the sample size, ℓ_1-MKL is significantly outperformed by the sum-kernel. On the contrary, non-sparse MKL significantly achieves higher AUC values than the ℓ_∞-MKL for sample sizes up to 20k. The scenario is well suited for ℓ_2-norm MKL which performs best. Finally, for 60k training instances, all methods but ℓ_1-norm MKL yield the same performance. Again, the superior performance of non-sparse MKL is remarkable, and of significance for the application domain: the method using the unweighted sum of kernels (Sonnenburg et al., 2006b) has recently been confirmed to be the 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 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

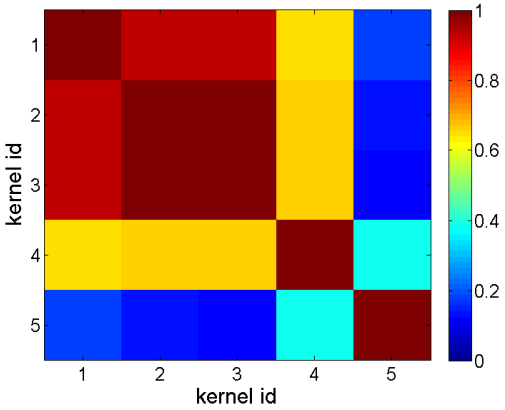

Figure 3: Pairwise alignments of the kernel matrices are shown for the gene start recognition experiment. From left to right, the ordering of the kernel matrices is TSS signal, promoter, 1st exon, angles, and energies. The first three kernels are highly correlated, as expected by their high AUC performances (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.

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. 2), 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.

To investigate this hypothesis we computed the pairwise alignments[12] of the centered kernel matrices, i.e., $\mathcal{A}(i,j) = \frac{<K_i,K_j>}{||K_i||_2||K_j||_2}$, with respect to the Frobenius dot product (eg., Golub and van Loan, 1996). The computed alignments are shown in Fig. 3. 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 the single kernel performance (AUC=0.74). We conjecture that the 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. 2(right), and the reason for the empirical success of non-sparse MKL on this data set.

5.4 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

12. The alignments can be interpreted as empirical estimates of the Pearson correlation of the kernels (Cristianini et al., 2002).

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.

	AUC ± 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	$\mathbf{83.13 \pm 0.6}$	

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 that $p = \infty$ 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 invidually. Their correlation is only moderate, as shown in Fig. 4, suggesting that they contain complementary information. Hence, downweighting one of the 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.

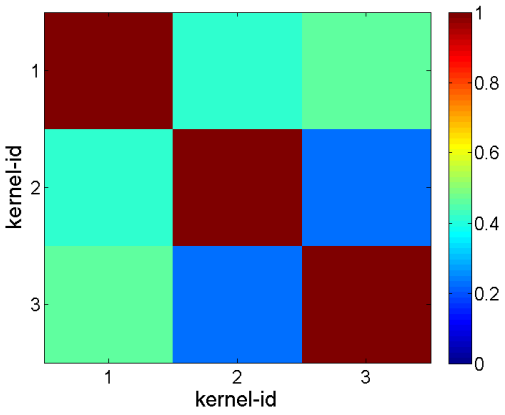

Figure 4: Pairwise alignments of the kernel matrices are shown for the metabolic gene network experiment. From left to right, the ordering of the kernel matrices is EXP, LOC, and PHY. One can see that all kernel matrices are equally correlated. Generally, the alignments are relatively low, suggesting that combining all kernels with equal weights is beneficial.

5.5 Execution Time

In this section we demonstrate the efficiency of our implementations of non-sparse MKL. We experiment on the MNIST data set[13], 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 ℓ_p-norm MKL with the state-of-the art for ℓ_1-norm MKL, namely simpleMKL[14] (Rakotomamonjy et al., 2008) and SILP-based wrapper and SILP-based chunking (Sonnenburg et al., 2006a). To this end, we perform 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*.

In addition, we solve standard SVMs[15] using the unweighted-sum kernel (ℓ_∞-norm MKL) as baseline. We optimize all methods up to a precision of 10^{-3} for the outer SVM-ε and 10^{-5} for the "inner" SIP precision and computed relative duality gaps. To provide a fair stopping criterion to simpleMKL, we set the stopping criterion of simpleMKL to the relative duality gap of its ℓ_1-norm counterpart. This way, the deviations of relative objective values of ℓ_1-norm MKL variants are guaranteed to be smaller than 10^{-4}. SVM trade-off parameters are set to $C = 1$ for all methods.

13. This data set is available from `http://yann.lecun.com/exdb/mnist/`.
14. We obtained an implementation from `http://asi.insa-rouen.fr/enseignants/~arakotom/code/`.
15. We use SVMlight as SVM-solver.

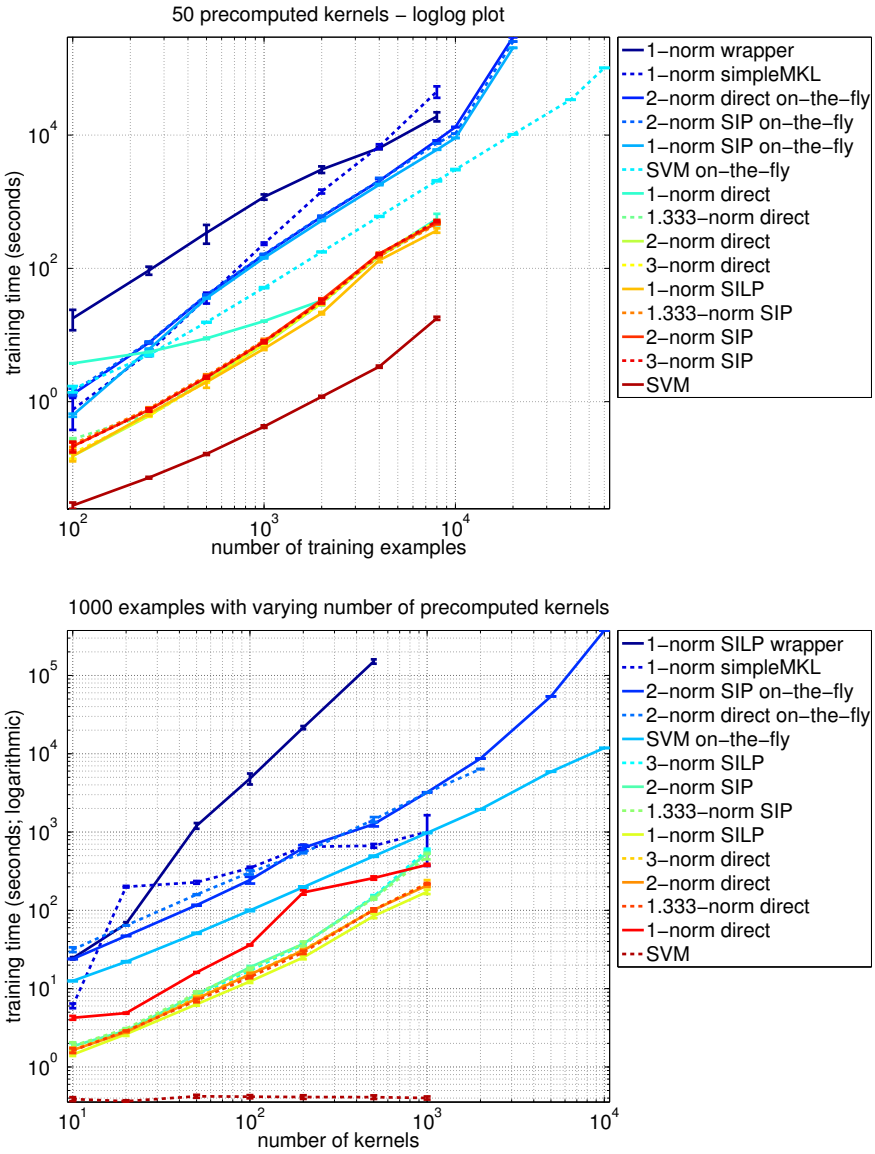

Figure 5: Execution times of SVM Training, ℓ_p-norm MKL based on interleaved optimization via direct optimization, the cutting plane algorithm (CPA) and the SimpleMKL wrapper, direct wrapper and CPA wrapper. (top) Training using fixed number of 50 kernels varying training set size. (bottom) For 1000 examples and varying numbers of kernels. Notice the tiny error bars and that these are log-log plots.

Scalability of the Algorithms w.r.t. Sample Size Figure 5 (top) displays the results for varying sample sizes and 50 precomputed or on-the-fly computed Gaussian kernels with bandwidths $2\sigma^2 \in 1.2^{0,\dots,49}$. 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 MKL wrapper based methods, SimpleMKL and the SILP wrapper, are the slowest; they are even slower than methods that compute kernels on-the-fly.

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). The reason is that 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 proposed optimization strategies the analytic optimization and the sequence of SIPs. Naturally, the generalized SIPs are slightly slower than the SILP variant (Section 4.2) since they solve an additional series of Taylor expansions within each $\boldsymbol{\theta}$-step.

Scalability of the Algorithms w.r.t. the Number of Kernels Figure 5 (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 $2\sigma^2 \in 1.2^{0,\dots,M-1}$. 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. 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).

Overall, our proposed interleaved analytic and cutting plane based optimization strategies achieve a speedup of up to two orders of magnitude over SimpleMKL. 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 than further accelerating the optimization algorithm.

6. Conclusion

We translated multiple kernel learning into a regularized risk minimization problem for arbitrary convex loss functions, Hilbertian regularizers, and arbitrary norm-penalties on

the mixing coefficients. Our general formulation can be motivated by both Tikhonov and Ivanov regularization approaches. Applied to previous MKL research, our result provides a unifying view and shows so far seemingly different MKL approaches to be equivalent.

Furthermore, we presented a general dual formulation of multiple kernel learning that subsumes many existing algorithms. We devised two efficient optimization schemes for non-sparse ℓ_p-norm MKL with $p \geq 1$: an analytic update for the mixing coefficients and a semi-infinite programming approach, both interleaved with chunking-based SVM training to allow for application at large scales. Our implementations are freely available and included in the SHOGUN toolbox. The execution times of our algorithms revealed that the interleaved optimization vastly outperforms commonly used wrapper approaches. Our results and the scalability of our MKL approach pave the way for other real-world applications of multiple kernel learning.

In order to empirically validate our ℓ_p-norm MKL model, we applied it to artificially generated data and real-world problems from computational biology. For the controlled toy experiment, where we simulated various levels of sparsity, ℓ_p-norm MKL achieved a low test error in all scenarios for scenario-wise tuned parameter p. Moreover, we studied three real-world problems showing that the choice of the norm is crucial for state-of-the art performance. For the TSS recognition, non-sparse MKL raised the bar in predictive performance, while for the other two tasks either sparse MKL or the unweighted-sum mixture performed best. In those cases the best solution can be arbitrarily closely approximated by ℓ_p-norm MKL with $1 < p < \infty$. Hence it seems natural that we observed non-sparse MKL to be never worse than an unweighted-sum kernel or a sparse MKL approach. Moreover, empirical evidence from our experiments along with others suggests that the popular ℓ_1-norm MKL is more prone to bad solutions than higher norms, despite appealing guarantees like the model selection consistency (Bach, 2008). A first step towards a learning-theoretical understanding of this empirical behaviour may be the convergence analysis of sparse estimators undertaken by Leeb and Pötscher (2008). However even restricted to ℓ_1-norm and ℓ_2-norm this issue is not yet resolved, and there is an apparent lack of theoretical underpinning of the general ℓ_p-norm case that yet remains to be filled.

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 solutions. 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. However, the exploration of theoretical explanations is beyond the scope of this submission. Nevertheless, we remark that even completely redundant but uncorrelated kernels may improve the predictive performance of a model, as averaging over several of them can reduce the variance of the predictions. Intuitively speaking, we observe clearly that in some cases all features, even though they may contain redundant information, should be kept, since putting their contributions to zero worsens prediction, i.e. all of them are informative to our MKL models.

Finally, we would like to note that it may be worthwhile to rethink the current strong preference for sparse models in the scientific community. A main reason for favoring sparsity may be the presumed interpretability of sparse models. This is not the topic and goal of

this article; however we remark that in general the identified model is sensitive to kernel normalization, and in particular in the presence of strongly correlated kernels, the results may be somewhat arbitrary, putting their interpretation in doubt. However, in the context of this work the predictive accuracy is of focal interest, and in this respect we demonstrate that non-sparse models may improve quite impressively over sparse ones.

Acknowledgments

The authors wish to thank Pavel Laskov, Motoaki Kawanabe, Vojtech Franc, Peter Gehler, Gunnar Rätsch, Peter Bartlett and Klaus-Robert Müller for fruitful discussions and helpful comments. This work was supported in part by the German Bundesministerium für Bildung und Forschung (BMBF) under the project REMIND (FKZ 01-IS07007A), by the German Academic Exchange Service, and by the FP7-ICT Programme 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.

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

$$\min_{\boldsymbol{x} \in D} \quad f(\boldsymbol{x}) + \sigma g(\boldsymbol{x}), \tag{34a}$$

$$\min_{\boldsymbol{x} \in D : g(\boldsymbol{x}) \leq \tau} \quad f(\boldsymbol{x}). \tag{34b}$$

Assume that the minima exist and that a constraint qualification holds in (34b)*, which gives rise to strong duality, e.g., that Slater's condition is satisfied. Furthermore assume that the constraint is active in the optimal point, i.e.*

$$\inf_{\boldsymbol{x} \in D} \; f(\boldsymbol{x}) \; < \inf_{\boldsymbol{x} \in D : g(\boldsymbol{x}) \leq \tau} \; f(\boldsymbol{x}). \tag{35}$$

Then we have that for each $\sigma > 0$ there exists a $\tau > 0$, and vice versa, such that OP (34a) *is equivalent to OP* (34b)*, i.e., each optimal solution of the one is an optimal solution of the other, and vice versa.*

Proof
(a). Let be $\sigma > 0$ and $\boldsymbol{x}^*$ be the optimal of (34a). We have to show that there exists a $\tau > 0$ such that $\boldsymbol{x}^*$ is optimal in (34b). We set $\tau = g(\boldsymbol{x}^*)$. Suppose $\boldsymbol{x}^*$ is not optimal in (34b), i.e., it exists $\tilde{\boldsymbol{x}} \in D : g(\tilde{\boldsymbol{x}}) \leq \tau$ such that $f(\tilde{\boldsymbol{x}}) < f(\boldsymbol{x}^*)$. Then we have

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

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

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

This contradics the optimality of $\boldsymbol{x}^*$ in (34a), and hence shows that $\boldsymbol{x}^*$ is optimal in (34b), which was to be shown.
(b). Vice versa, let $\tau > 0$ be $\boldsymbol{x}^*$ optimal in (34b). The Lagrangian of (34b) is given by

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

By strong duality $\boldsymbol{x}^*$ is optimal in the sattle point problem

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

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

$$\min_{\boldsymbol{x} \in D} \quad f(\boldsymbol{x}) + \sigma^* \left(g(\boldsymbol{x}) - \tau\right). \tag{36}$$

Removing the constant term $-\sigma^*\tau$, and setting $\sigma = \sigma^*$, we have that $\boldsymbol{x}^*$ is optimal in (34a), which was to be shown. Moreover by (35) we have that

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

and hence we see from Eq. (36) that $\sigma^* > 0$, which completes the proof of the proposition. ■

References

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

J. Aflalo, A. Ben-Tal, C. Bhattacharyya, J. S. Nath, and S. Raman. Variable sparsity kernel learning — algorithms and applications. *Journal of Machine Learning Research*, 2010. Submitted 12/2009.

M. Anitescu. A superlinearly convergent sequential quadratically constrained quadratic programming algorithm for degenerate nonlinear programming. *SIAM J. on Optimization*, 12(4):949–978, 2002.

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. Consistency of the group lasso and multiple kernel learning. *J. Mach. Learn. Res.*, 9:1179–1225, 2008.

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.

A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167 – 175, 2003.

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

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, 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, 2009.

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

R. Fan, K. Chang, C. Hsieh, X. Wang, and C. 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 svm. *Journal of Machine Learning Research*, 6:1889–1918, 2005.

D. Fernández and M. Solodov. On local convergence of sequential quadratically-constrained quadratic-programming type methods, with an extension to variational problems. *Comput. Optim. Appl.*, 39(2):143–160, 2008.

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. URL `http://ida.first.fraunhofer.de/~franc/ocas/html/index.html`.

M. Fukushima, Z.-Q. Luo, and P. Tseng. A sequential quadratically constrained quadratic programming method for differentiable convex minimization. *SIAM J. on Optimization*, 13(4):1098–1119, 2002.

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

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

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

V. Ivanov, V. Vasin, and V. 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. Burges, and A. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 169–184, Cambridge, MA, 1999. MIT Press.

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. Düssel, C. Gehl, and P. Laskov. Automatic feature selection for anomaly detection. In D. Balfanz and J. Staddon, editors, *AISec*, pages 71–76. ACM, 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.

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.

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

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

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, 2005a.

C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *JMLR*, 6: 1099–1125, 2005b.

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. Nakajima, A. Binder, C. Müller, W. Wojcikiewicz, M. Kloft, U. Brefeld, K.-R. Müller, and M. Kawanabe. Multiple kernel learning for object classification. In *Proceedings of the 12th Workshop on Information-based Induction Sciences*, 2009.

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. 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. Burges, and A. 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. 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. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

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

B. Schölkopf, S. Mika, C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. 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. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

M. V. Solodov. On the sequential quadratically constrained quadratic programming methods. *Math. Oper. Res.*, 29(1):64–79, 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.

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.

D. Tax and R. 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.

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.

F. Yan, K. Mikolajczyk, J. Kittler, and M. Tahir. A comparison of l1 norm and l2 norm multiple kernel svms in image and video classification. *International Workshop on Content-Based Multimedia Indexing*, 0:7–12, 2009.

S. Yu, T. Falck, A. Daemen, J. Suykens, B. D. Moor, and Y. Moreau. Non-sparse kernel fusion and its applications in genomic data integration. Technical report, KU Leuven, Nederland, 2009.

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.