ℓ_p-Norm Multiple Kernel Learning

Making Learning with Multiple Kernels Effective

ℓ_p-Norm Multiple Kernel Learning

vorgelegt von Dipl.-Math.

Marius Kloft

Von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
– Dr. rer. nat. –

genehmigte

Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Manfred Opper
Fakultät IV – Elektrotechnik und Informatik
Technische Universität Berlin

Berichter: Prof. Dr. Klaus-Robert Müller
Fakultät IV – Elektrotechnik und Informatik
Technische Universität Berlin

Prof. Dr. Peter L. Bartlett
Department of EECS and Department of Statistics
University of California, Berkeley, USA

Prof. Dr. Gilles Blanchard
Institut für Mathematik
Universität Potsdam

Tag der mündlichen Aussprache: 26.09.2011

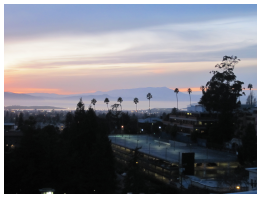
Berlin, 2011
D 83

To my parents

Acknowledgements

I am deeply grateful for the opportunity to have worked with so many brilliant and creative minds over the last years. First of all, I would like to thank my PhD advisors Prof. Dr. Klaus-Robert Müller, Prof. Dr. Peter L. Bartlett, and Prof. Dr. Gilles Blanchard. Of course, I owe the utmost gratitude to Professor Müller, who initially introduced me to the subject of machine learning in 2007; with his infectious optimism, wit, and curiosity, he created the open and stimulating atmosphere that characterizes his Machine Learning Laboratory at TU Berlin. It was also Professor Müller who gave me advice and support in various respects from the very beginning of my PhD studies, thus turning out to be much more than just a scientific advisor. I am deeply thankful to him. By the same token, I would like to thank all members of his lab for creating such a pleasurable working atmosphere in the office each day, most notably my former and current office mates Pascal Lehwark, Nico Görnitz, Gregoire Montavon, and Stefan Haufe.

Furthermore, I would like to thank Professor Bartlett very much indeed for kindly inviting me over to visit his learning theory group at UC Berkeley, a most enjoyable and inspiring experience in many ways, and, most of all, for introducing me to the fascinating world of learning theory; his readiness to share his vast knowledge with me allowed me to gain a deeper understanding of the field. I also thank all members of his group, especially Alekh, for our stimulating discussions, not limited to various mathematical subjects, but also in matters of the everyday life. I also thank all office mates in Sutardja Dai Hall for the nice atmosphere, and the UC Berkeley for providing such a great office with an absolutely stunning view of the SF Bay, all of which contributed to making my stay (Oct 2009–Sep 2010) such a memorable one.



Likewise, I am very grateful to Professor Blanchard for taking so much of his valuable time for our extensive and fruitful discussions on various mathematical problems, sharing his immense knowledge and deep insights with me. I remember us sitting in cafés for hours, trying to solve a mathematical puzzle arising from our joint research

projects. It is rare that a professor invests so much of his time in mentoring a student and I owe him a big thanks!

My special thanks go to Dr. Ulf Brefeld for his caring mentoring and patient teaching in the initial phase of my PhD, which considerably eased the start. His encouragement gave me strength and confidence. Likewise, I thank Dr. Ulrich Rückert for his mentoring while both of us were with the UC Berkeley, which substantially sped up my introduction to learning theory. Moreover, I am indebted to Dr. Alexander Zien for countless stimulating discussions and to Dr. Sören Sonnenburg and Alexander Binder for sharing their great insights in the SHOGUN toolbox with me. Furthermore, I would like to thank all members of the REMIND project team, especially Dr. Konrad Rieck and Dr. Pavel Laskov, for the warm and nice atmosphere that made working in the team (2007–2009) such an effective and also fun experience.

I especially thank Claudia for so carefully proof-reading my manuscript in the final stage with respect to grammar and spelling, and Helen, who suffered the most from my obsession with unfinished chapters and last-minute changes to the manuscript, for her patience and support. Most of all, I would like to thank my parents for supporting me in every conceivable way.

Finally, I acknowledge financial support of the German Bundesministerium für Bildung und Forschung (BMBF), under the project REMIND (FKZ 01-IS07007A), and the European Community, under the PASCAL2 Network of Excellence of the FP7-ICT program (ICT-216886). I acknowledge the Berlin Institute of Technology (TU Berlin) and the German Academic Exchange Service (DAAD) for PhD student scholarships of 1-year runtime each.

Abstract

The goal of machine learning is to learn unknown concepts from data. In real-world applications such as bioinformatics and computer vision, data frequently arises from multiple heterogeneous sources or is represented by various complementary views, the right choice—or even combination—of which being unknown. To this end, the multiple kernel learning (MKL) framework provides a mathematically sound solution. Previous approaches to learning with multiple kernels promote sparse kernel combinations to support interpretability and scalability. Unfortunately, classical approaches to learning with multiple kernels are rarely observed to outperform trivial baselines in practical applications.

In this thesis, I approach learning with multiple kernels from a unifying view which shows previous works to be only particular instances of a much more general family of multi-kernel methods. To allow for more effective kernel mixtures, I have developed the *ℓ_p-norm multiple kernel learning* methodology, which, to sum it up, is both more efficient and more accurate than previous approaches to multiple kernel learning, as demonstrated on several data sets. In particular, I derive optimization algorithms that are much faster than the commonly used ones, allowing to deal with up to ten thousands of data points and thousands of kernels at the same time. Empirical applications of ℓ_p-norm MKL to diverse, challenging problems from the domains of bioinformatics and computer vision show that ℓ_p-norm MKL achieves accuracies that surpass the state-of-the-art.

The proposed techniques are underpinned by deep foundations in the theory of learning: I prove tight lower and upper bounds on the local and global Rademacher complexities of the hypothesis class associated with ℓ_p-norm MKL, which yields excess risk bounds with fast convergence rates, thus being tighter than existing bounds for MKL, which only achieve slow convergence rates. I also connect the minimal values of the bounds with the soft sparsity of the underlying Bayes hypothesis, proving that for a large range of learning scenarios ℓ_p-norm MKL attains substantial stronger generalization guarantees than classical approaches to learning with multiple kernels. Using a methodology based on the theoretical bounds, and exemplified by means of a controlled toy experiment, I investigate *why* MKL is effective in real applications.

Data sets, source code and implementations of the algorithms, additional scripts for model selection, and further information are freely available online.

Zusammenfassung

Ziel des Maschinellen Lernens ist das Erlernen unbekannter Konzepte aus Daten. In vielen aktuellen Anwendungsbereichen des Maschinellen Lernens, wie zum Beispiel der Bioinformatik oder der Computer Vision, sind die Daten auf vielfältige Art und Weise in Merkmalsgruppierungen repräsentiert. Im Voraus ist allerdings die optimale Kombination jener Merkmalsgruppen oftmals unbekannt. Die Methodologie des Lernens mit mehreren Kernen bietet einen attraktiven und mathematisch fundierten Ansatz zu diesem Problem. Existierende Modelle konzentrieren sich auf dünn besetzte Merkmals- bzw. Kernkombinationen, um deren Interpretierbarkeit zu erleichtern. Allerdings erweisen sich solche klassischen Ansätze zum Lernen mit mehreren Kernen in der Praxis als wenig effektiv.

In der vorliegenden Dissertation betrachte ich das Problem des Lernens mit mehreren Kernen aus einer neuartigen, generelleren Perspektive. In dieser Sichtweise sind klassische Ansätze nur Spezialfälle eines wesentlich generelleren Systems des Lernens mit mehreren Kernen. Um effektivere Kernmischungen zu erhalten, entwickle ich die ℓ_p-*norm multiple kernel learning* Methodologie, die sich effizienter und effektiver als vorherige Lösungsansätze erweist. Insbesondere leite ich Algorithmen zur Optimierung des Problems her, die wesentlich schneller sind als existierende und es erlauben, gleichzeitig Zehntausende von Trainingsbeispielen und Tausende von Kernen zu verarbeiten. Ich analysiere die Effektivität unserer Methodologie in einer Vielzahl von schwierigen und hochzentralen Problemen aus den Bereichen Bioinformatik und Computer Vision und zeige, dass ℓ_p-*norm multiple kernel learning* Vorhersagegenauigkeiten erreicht, die den neuesten Stand der Forschung übertreffen.

Die entwickelten Techniken sind tief untermauert in der Theorie des Maschinellen Lernens: Ich beweise untere und obere Schranken auf die Komplexität der zugehörigen Hypothesenklasse, was die Herleitung von Generalisierungsschranken erlaubt, die eine schnellere Konvergenzgeschwindigkeit haben als vorherige Schranken. Des Weiteren stelle ich den minimalen Wert der Schranken mit den geometrischen Eigenschaften der Bayes-Hypothese in Verbindung. Darauf basierend beweise ich, dass für eine große Anzahl von Szenarien ℓ_p-*norm multiple kernel learning* deutlich stärkere Generalisierungsgarantien aufweist als vorherige Ansätze zum Lernen mit mehreren Kernen. Mit Hilfe einer von mir vorgeschlagenen Methodik, basierend auf den theoretischen Schranken und sogenannten *kernel alignments*, untersuche ich, warum sich ℓ_p-*norm multiple kernel learning* als hocheffektiv in praktischen Anwendungsgebieten erweist.

Die eingesetzten Datensätze, der Quellcode und die Implementierungen der Algorithmen sowie weitere Informationen zur Benutzung sind online frei verfügbar.

Contents

Part I

Introduction and Overview

I Introduction and Overview

1 Introduction

The goal of machine learning is to learn unknown concepts from data. But the success of a learning machine crucially depends on the quality of the data representation. At this point, the paradigm of kernel-based learning (Schölkopf et al., 1998; Müller et al., 2001) offers an elegant way for decoupling the learning and data representation processes in a modular fashion. This allows to obtain complex learning machines from simple linear ones in a canonical way. Nowadays, kernel machines are frequently employed in modern application domains that are characterized by vast amounts of data along with highly non-trivial learning tasks such as bioinformatics or computer vision, for their favorable generalization performance while maintaining computational feasibility.

However, after more than a decade of research it still remains an unsolved problem to find the best *kernel* for a task at hand. Most frequently, the kernel is selected from a candidate set according to its generalization performance on a validation set, which is held back at training time. Clearly, the performance of such an algorithm is limited by the performance of the best kernel in the set and can be arbitrarily bad if the kernel does not match the underlying learning task. Unfortunately, in the current state of research, there is little hope that in the near future a machine will be able to automatically engineer the *perfect* kernel for a particular problem at hand (Searle, 1980). However, by restricting ourselves to a less general problem, can we legitimately hope to obtain a mathematically sound solution? And if so, which restrictions have to be imposed?

A first step towards a more realistic model of learning the kernel was achieved in Lanckriet et al. (2004a), who showed that, given a candidate set of kernels, it is computationally feasible to simultaneously learn a support vector machine *and* a linear kernel combination at the same time, if the so-formed kernel combinations are required to be positive-definite and trace-norm normalized. This framework was entitled *multiple kernel learning* (MKL). Research in the following years focused on speeding up the initially demanding optimization algorithms (e.g. Sonnenburg et al., 2006a; Rakotomamonjy et al., 2008)—ignoring the fact that empirical evidence for the superiority of learning with multiple kernels over single-kernel baselines was missing.

By imposing an ℓ_1-norm regularizer on the kernel weights, classical approaches to multiple kernel learning promote *sparse* kernel combinations to support interpretability and scalability. Unfortunately, sparseness is not always beneficial and can be restrictive in practice, for example, in the presence of complementary kernel sets. However, nega-

tive results are less often published in science than positive ones. It took until 2008 for concerns regarding the effectiveness of multiple kernel learning in practical applications to be raised, starting in the domains of bioinformatics (Noble, 2008) and computer vision (Gehler and Nowozin, 2008): a multitude of researchers presented empirical evidence showing that, in practice, multiple kernel learning is frequently outperformed by a simple uniform kernel combination (Cortes et al., 2008, 2009a; Gehler and Nowozin, 2009; Yu et al., 2010). The whole discussion peaked in the provocative question "Can learning kernels help performance?" posed by Corinna Cortes in an invited talk at ICML 2009 (Cortes, 2009).

Consequently, despite all the substantial progress in the field of multiple kernel learning, 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). Even worse, despite the recent attempts for clarification (Lanckriet et al., 2009), underlying reasons for the empirical picture remain unclear. At this point, I argue that all of this is now achievable, thus answering Corinna Cortes's research question in the affirmative:

1.1 Author's PhD Thesis

This dissertation concerns the validation of the following thesis:

> *ℓ_p-norm multiple kernel learning, a methodology that I developed and of which I show that it enjoys favorable theoretical guarantees, is both faster and more accurate than existing approaches to learning with multiple kernels, finally making multiple kernel learning effective in practical applications.*

1.2 Organization of this Dissertation and Own Contributions

Part I: Introduction and Overview

I start my dissertation in Chapter 1 with a short introduction to and motivation of multiple kernel learning (MKL), containing, in a nutshell, the statement of the problem to be solved and examples of practical applications where it arises, taken from the domains of bioinformatics and computer vision.

In Chapter 2, I formally introduce a rigorous mathematical view of the problem, deferring mathematical preliminaries to Appendix A. Deviating from standard introductions, I phrase MKL as a general optimization criterion based on structured regularization, covering and also unifying existing formulations under a common umbrella; from this point of view, classical MKL is only a particular instance of a more general family of MKL methods. This allows to analyze a large variety of MKL methods jointly, as exemplified by deriving a general dual representation of the criterion,

without making assumptions on the employed norm or the loss, beside being convex. This not only delivers insights into connections between existing MKL formulations, but also allows to derive new ones as special cases of the unifying view.

Previously Published Work

This thesis is based on the following *selected* publications.

The core framework was published in:

> [1] **M. Kloft**, U. Brefeld, P. Laskov, and S. Sonnenburg. Non-sparse multiple kernel learning. In *Proc. of the NIPS 2008 Workshop on Kernel Learning: Automatic Selection of Kernels*, 2008.
>
> [2] **M. Kloft**, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Müller, and A. Zien. Efficient and accurate lp-norm multiple kernel learning. In *Advances in Neural Information Processing Systems 22* (NIPS 2009), pages 997–1005. MIT Press, 2009.
>
> [3] **M. Kloft**, U. Brefeld, S. Sonnenburg, and A. Zien. Lp-norm multiple kernel learning. *Journal of Machine Learning Research* (JMLR), 12:953–997, 2011.

The main idea of ℓ_p-norm MKL was initially presented in [1]. The core framework was subsequently published in [2]-[3].

Theoretical aspects were presented in:

> [4] **M. Kloft**, U. Rückert, and P. L. Bartlett. A unifying view of multiple kernel learning. In *Proceedings of the European Conference on Machine Learning* (ECML), 2010.
>
> [5] **M. Kloft** and G. Blanchard. The local Rademacher Complexity of Multiple Kernel Learning. *ArXiv preprint 1103.0790v1, 2011.* Short version submitted to NIPS 2011, Jun 2011. Full version submitted to *Journal of Machine Learning Research* (JMLR), Mar 2011.

Applications to computer vision were discussed in:

> [6] A. Binder, S. Nakajima, **M. Kloft**, C. Müller, W. Wojcikiewicz, U. Brefeld, K.-R. Müller, and M. Kawanabe. Classifying Visual Objects with Multiple Kernels. Submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. A preliminary version is published in *Proceedings of the 12th Workshop on Information-based Induction Sciences*, 2010.

The content of this thesis is related to the above publications in the following way. Chapter 2 is based on [4]; Chapter 3 is based on [1]-[3]; Chapter 4 is based on [5]; Chapter 5 contains material from [1]-[3] and [6].

> NOTE*: a* complete *list of all publications is shown at the end of the bibliography section.*

Part II: ℓ_p-norm Multiple Kernel Learning

In the main part of my dissertation, I introduce a novel instantiation of the proposed general criterion, which I entitle *ℓ_p-norm multiple kernel learning*. Recognizing classical approaches to learning with multiple kernels as the special case of deploying ℓ_1-norm and hinge loss, I argue that, from the structured point of view, it is more natural to chose an intermediate norm rather than an extreme ℓ_1-norm. I show the general connection of the structured formulation to the learning-kernels formulation that is usually considered in the literature. For classical MKL, this connection is known from the seminal paper of Bach et al. (2004); here, I show that it applies to a whole family of MKL algorithms, no matter which convex loss function or structured ℓ_p-norm regularizer is employed. The remainder of my dissertation focuses on the analysis of ℓ_p-norm MKL in terms of optimization algorithms, theoretical justification, empirical analysis, and applications to bioinformatics and computer vision.

Chapter 3 is on optimization algorithms. Considering the gain in prediction accuracy achieved by ℓ_p-norm MKL, established later in this dissertation, one might expect a substantial drawback with respect to execution time—the contrary is the case: the presented algorithms allow us to deal with ten thousands of data points and thousands of kernels at the same time, being up to two magnitudes faster than the state-of-the-art in MKL research, including HessianMKL and SimpleMKL. Some of these, like the cutting-plane strategy, which was designed by me and implemented with the help of Sören Sonnenburg and Alexander Zien, are based on previous work by Sonnenburg et al. (2006a), others like the analytical solver are completely novel. The latter one is based on a simple analytical formula that can be evaluated in micro seconds, and thus, despite its efficiency, is even simpler than SimpleMKL, which requires a heuristic line search. I show it being provably convergent, using the usual regularity assumptions. I also wrote macro scripts, completely automating the whole process from training over model selection to evaluation. Currently, MKL can be trained and validated by a single line of code including random subsampling, model search for the optimal parameters C and p, and collection of results.[1]

In Chapter 4, the proposed techniques are justified from a theoretical point of view. I prove tight lower and upper bounds on the local and global Rademacher complexities of the hypothesis class associated with ℓ_p-norm MKL, which yields excess risk bounds with fast convergence rates, thus being tighter than existing bounds for MKL, which only achieved slow convergence rates. For the results on the local complexities to hold, I find an assumption on the uncorrelatedness of the kernels; a similar assumption was also recently used by Raskutti et al. (2010), but in the different context of sparse recovery.

Even the tightest previous theoretical analyses such as the one carried out by Cortes et al. (2010a) for the special case of classical MKL were not able to answer the research question "Can learning kernels help performance?" (Cortes, 2009). In contrast, beside

[1] Implementation freely available under the GPL license at `http://doc.ml.tu-berlin.de/nonsparse_mkl/`.

reporting on the worst-case bounds, I also connect the minimal values of the theoretical bounds with the geometry of the underlying learning scenario (namely, the soft sparsity of the Bayes hypothesis), in particular, proving that for a large range of learning scenarios ℓ_p-norm MKL attains a strictly "better" (i.e., lower) bound than classical ℓ_1-norm MKL and the SVM using a uniform kernel combination. This theoretically justifies using ℓ_p-norm MKL and multiple kernel learning in general.

Chapter 5 concerns the empirical analysis of ℓ_p-norm MKL and applications to diverse, challenging problems from the domains of bioinformatics and computer vision. From a practical point of view, this is the most important chapter of my dissertation as I show here that ℓ_p-norm MKL works well in practice. For the experiments, problems from the domains of bioinformatics and computer vision were chosen, not only because they come with highly topical, challenging, small- and large-scale prediction tasks, but also because researchers frequently encounter multiple kernels or data sources here. This renders these domains especially appealing for the use of MKL.

At this point, it has to be admitted that other researchers also deployed MKL to those domains: Lanckriet et al. (2004b) experimented on bioinformatics data and Varma and Ray (2007) and Gehler and Nowozin (2009) on computer vision data. However, none of those studies were able to prove the practical effectiveness of MKL. The first study investigated whether MKL can help performance in genomic data fusion: indeed, MKL outperformed the best single-kernel SVM as determined by model selection; however, the uniform kernel combination was not investigated at that point. Subsequent investigations showed here that the latter outperforms MKL on the very same data set.[2] In the second study, MKL was shown to substantially outperform the uniform kernel combination on the caltech-101 object recogniction data set. This study turned out to be incorrect due to a flaw in the kernel generation.[3] Subsequently, MKL was studied on the very same data set by Gehler and Nowozin (2009) and found to be outperformed by an SVM using a uniform kernel combination. To the best of my knowledge, the only confirmed experiment concerning MKL outperforming the SVM using a uniform kernel combination is the one undertaken by Zien and Ong (2007) in the context of protein subcellular localization prediction.

In this thesis, I show that by considering ℓ_p-norms, MKL can in fact help performance in both of the above applications (genomic data fusion and object recognition). Besides, I study applications to gene transcription splice site detection, protein fold prediction, and metabolic network reconstruction. While I observe that MKL helps performance in some applications (including the ones mentioned in the above paragraph, where researchers tried for years making MKL effective), I also show that sometimes MKL does *not* increase the performance (this is, for example, the case for the metabolic network reconstruction experiment). The raises the question *why* it sometimes helps and why sometimes it does not. At this point, I introduce a methodology deploying both, the

[2] Personal correspondences with William S. Noble; see W. Noble's talk at `http://videolectures.net/lkasok08_whistler/`, June 20, 2011.

[3] See errata on the first author's personal homepage, `http://research.microsoft.com/en-us/um/people/manik/projects/trade-off/caltech101.html`, June 20, 2011.

bounds that I prove in the theoretical chapter of this thesis and the kernel alignment techniques initially proposed in a different context by Cristianini et al. (2002). While the theoretical bounds are used to investigate the optimal norm parameter p, showing that the effectiveness of MKL is connected with the soft sparsity of the underlying Bayes hypothesis, the alignments are used to study whether the kernels at hand are complementary or rather redundant. The whole methodology is exemplified by means of a toy experiment, where I artificially construct the Bayes hypothesis, controlling the underlying soft sparsity of the problem. It is shown that MKL's empirical performance can crucially depend on the choice of the norm parameter p and that the optimality of such a parameter can highly depend on the geometry of the underlying Bayes hypothesis (it can make the difference between 4% or 43% test error, as shown in the simulations). The chapter concludes with my study, carried out with the help of Shinichi Nakajima, on object recogniction, the very same application unsuccessfully studied earlier by Varma and Ray (2007) and Gehler and Nowozin (2009)—see discussion above—where classical MKL could not help performance. In contrast, I show that, by deploying the proposed *ℓ_p-norm multiple kernel learning* and taking $p = 1.11$ in median, the prediction accuracy can be raised over the SVM baselines, regardless of the class, by an AP score of 1.5 in average, and for 7 out of the 20 classes significantly so, concluding the final Chapter 5 of my thesis.

1.3 Multiple Kernel Learning in a Nutshell

In this section, we introduce the problem of multiple kernel learning.

Problem setting In classical supervised machine learning we are given training examples $x_1, \ldots, x_n$ lying in some input space $\mathcal{X}$ and labels $y_1, \ldots, y_n \in \mathcal{Y}$, in the simplest case, $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{-1, 1\}$. The goal in supervised learning is to find a prediction function $f : \mathcal{X} \to \mathcal{Y}$ from a given set $H \subset \mathcal{Y}^{\mathcal{X}}$ that has a low error rate on new data $\big(x_{n+1}, y_{n+1}\big), \ldots, \big(x_{n+l}, y_{n+l}\big)$ stemming from the same data source but which is unseen at training time. Clearly, we cannot learn anything useful at all if the training data and the new data are not connected in any way. Therefore one usually assumes a stochastic mechanism underlying the data generation process, most commonly, that all of the $\big(x_i, y_i\big)$ are drawn independently from one and the same probability distribution P. In this case the quality of the prediction function f is measured by the expected rate of false predictions $\mathbb{E}_{(x,y)\sim P} \mathbb{1}_{\{f(x)\neq y\}}$.

Regularized risk minimization An often-employed approach to this problem is *regularized risk minimization* (RRM), where a minimizer

$$f^* \in \operatorname{argmin}_{f \in H} \; \Omega(f) + C L_n(f)$$

is found. Hereby $L_n(f) = \sum_{i=1}^n l\left(f(x_i), y_i\right)$ is the (cumulative) *empirical loss* of a hypothesis f with respect to a function $l : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$ (called *loss function*) that upper bounds the 0-1 loss $\mathbb{1}_{\{f(x)\neq y\}}$ and $\Omega : H \to \mathbb{R}$ is a mapping (called *regularizer*). The name *risk* minimization stems from the fact that L_n is n-times the empirical risk. We can interpret RRM as minimizing a trade-off between the empirical loss (to classify the

training data well) and a regularizer (to penalize the complexity of f and thus avoid overfitting), where the trade-off is controlled by a positive parameter C.

Kernel methods Single-kernel approaches to RRM (Vapnik, 1998) simply use linear models of the form $f_{w,b}(x) = \langle w, x\rangle + b$ but—this is the core idea of kernel methods—replace all resulting inner products $\langle x, \bar{x}\rangle$ by so-called *kernels* $k(x, \bar{x})$. Roughly speaking, a kernel k is a clever way to efficiently compute inner products in a possibly very high-dimensional feature space. As outlined in the introduction, the ultimate goal of kernel learning would be to find the best kernel for a problem at hand—but this is a task too hard to allow for a general solution so that, in practice, the kernel is usually either fixed a priori or selected from a small candidate set $\{k_1, \dots, k_M\}$ according to its prediction error on a validation set, which is held back at training time.

Multiple kernel learning A first step towards finding an optimal kernel is multiple kernel learning. Here, instead of just picking a kernel from a set, a new kernel k is *constructed* by combining kernels from a possibly large given set. For example, the following combination rules give rise to valid kernel:

$$k = \theta_1 k_1 + \cdots + \theta_M k_M \qquad \text{(sums)}$$

$$k = k_1^{\theta_1} \cdot \ldots \cdot k_M^{\theta_M}\,, \qquad \text{(products)}$$

where $\boldsymbol{\theta} \in \mathbb{R}_+^M$. By searching for an optimal $\boldsymbol{\theta}$, we traverse an infinitely large set of "combined kernels". Unfortunately, except for very small M (typically $M \leq 3$), the search space is too large to be traversed by standard methods such as grid search.

The core idea of multiple kernel learning is based on the insight that most machine learning problems are formulated as solutions of optimization problems. What if we include the parameter $\boldsymbol{\theta}$ as a variable into the optimization? For example, for the support vector machine (SVM) this task becomes

$$\min_{\boldsymbol{\theta} \geq \mathbf{0}} \ \mathrm{SVM}\Big(\sum_{m=1}^{M} \theta_m k_m \Big)\,.$$

A first difficulty we face is that, without any further restriction on $\boldsymbol{\theta}$, the optimization problem may be unbounded or, for example, yield a trivial solution that does not generalize to new, unseen data. In the past, this has been addressed by restricting the search space to convex combinations (i.e., sums that add up to one). In that case the above problem becomes (Lanckriet et al., 2004a; Bach et al., 2004)

$$\min_{\boldsymbol{\theta} \geq \mathbf{0},\, \|\boldsymbol{\theta}\|_1 = 1} \ \mathrm{SVM}\Big(\sum_{m=1}^{M} \theta_m k_m \Big)\,.$$

For quite some time, the above regularization strategy was the prevalent one in multiple kernel learning research.

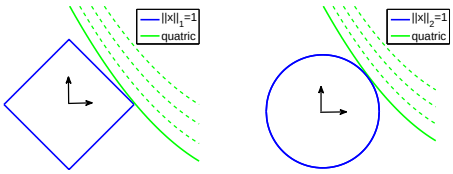
ℓ_p-Norm Multiple Kernel Learning Although having been folklore among researchers for quite a while already, it took until 2008 that criticism was made public concerning

the usefulness of the above approach in practical applications (see citations in the introduction): it is frequently observed being outperformed by a simple, regular SVM using a uniform kernel combination.

In this thesis, we propose to discard the restrictive convex combination requirement (which corresponds to using an ℓ_1-norm regularizer on $\boldsymbol{\theta}$) and use a more flexible ℓ_p-norm regularizer instead, leading to the optimization problem

$$\min_{\boldsymbol{\theta}\geq\mathbf{0},\, \|\boldsymbol{\theta}\|_p=1} \; \mathrm{SVM}\Big(\sum_{m=1}^{M} \theta_m k_m \Big) .$$

The difference between the two ways of regularizing is illustrated in the following figure, where the ℓ_1- and ℓ_2-norm regularized problems are compared. The blue line shows the norm constraint and the green one the level sets of a quadratic function. The optimal solution of the optimization problem is attained where the level sets of the objective function touch the norm constraint. If the objective function is convex (illustrated here for a quadratic function), and for ℓ_1-norm the point of intersection is likely to be at one of the corners of the square (shown on the figure to the left). Because the corners are likely to have zero-entries, sparsity is to be expected. This is in contrast to ℓ_2-norm regularization, where, in general, a non-sparse solution is to be expected (shown on the figure to the right).



Alternative view of MKL Another way to view MKL is based on the insight that a kernel k gives rise to a (possibly high-dimensional) feature map ϕ so that We can illustrate single kernel learning by the following diagram:

$$\text{input } x \overset{\text{kernel feature map}}{\longrightarrow} \phi(x) \overset{\text{linear discrimination}}{\longrightarrow} \langle \boldsymbol{w}, \phi(x) \rangle .$$

Correspondingly, when we are given multiple kernels $k_1, \ldots, k_M$, we also obtain multiple feature maps $\phi_1, \ldots, \phi_M$, one for each kernel. Thus the combined kernel $k = \sum_{m=1}^{M} \theta_m k_m$ corresponds to a "combined" feature map $\phi_{\boldsymbol{\theta}} = \sqrt{\theta_1} \times \cdots \times \sqrt{\theta_M}\phi_M$. This is illustrated in Figure 1.1.

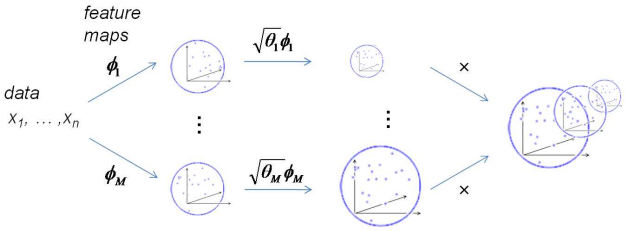


Figure 1.1: Illustration of multiple kernel learning in terms of weighted kernel feature spaces.

Examples In real-world applications such as bioinformatics and computer vision, data either frequently arises from multiple heterogeneous sources, describing different properties of one and the same object, or is represented by various complementary views, the right choice—or even combination—of which being unknown. In this case, multiple kernel learning (MKL) is especially appealing as it provides a mathematically sound solution to the data fusion problem.

For example, in *transcription splice site detection*, a bioformatics application, we can describe the properties of DNA by its twistedness, its binding energy, the position of the first exon, or the abstract string information obtained from the sequence of nucleotides. Each characterization gives rise to a different kernel: the energy kernel, the angle kernel, the first-exon kernel, and the string kernel. In Section 5.3 we show that this application highly profits from employing MKL. By training our large-scale implementation of ℓ_p-norm multiple kernel learning on up to 60,000 training examples and testing on 20,000 data points, we show that ℓ_p-norm MKL can significantly increase the prediction accuracy ℓ_1-norm MKL and the SVM using a uniform kernel combination. This is remarkable since the latter was recently confirmed to be the winner of an international comparison of 19 splice site detectors.

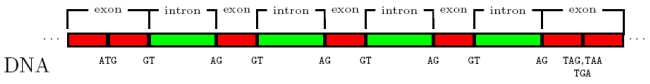


Figure 1.2: Figure taken from Sonnenburg (2008).

Another example is *object recognition*, a computer vision application. Here, the goal is to categorize images according to their information content, e.g., what kind of animal is shown in an image. Usually, the image is represented as a vector in some feature space; however, representations can be built from various features, for example, color, texture, and shape information. Clearly, there is no representation that is optimal for all tasks simultaneously. For example, color information is essential for the detection of stop signs in images but it is superfluous for finding cars. In this work we propose to let the learner figure out an optimal combination of features for the task at hand. In Section 5.5, we report on results of the well-known VOC 2008 challenge data set and show that the proposed ℓ_p-norm MKL achieves higher prediction accuracies than both, classical MKL and the SVM baseline.

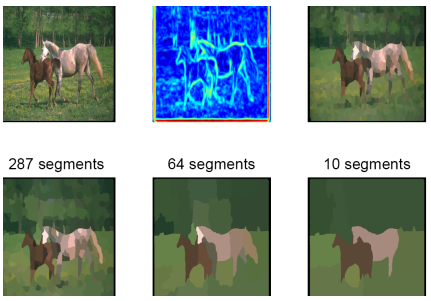


Figure 1.3: Figure taken from Bach (2008a).

1.4 Basic Notation

In this thesis, vectors are understood as column vectors and marked with boldface letters or symbols. However, for structured elements

$$\boldsymbol{w} = \left(\boldsymbol{w}_1^\top, \ldots, \boldsymbol{w}_M^\top\right)^\top \in \mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_M},$$

in slight deviation to this notation, the simpler expression $\boldsymbol{w} = \left(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M\right)$ is used. Likewise, the notation $(\boldsymbol{u}^{(m)})_{m=1}^M$ for the element $\boldsymbol{u} = \left(\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(M)}\right) \in \mathcal{H} = \mathcal{H}_1 \times \ldots \times \mathcal{H}_M$, where $\mathcal{H}, \mathcal{H}_1, \ldots, \mathcal{H}_M$ are Hilbert spaces, is frequently used.

Vectors in $\mathbb{R}^n$ of all zeros or ones are denoted by $\mathbf{0}$ and $\mathbf{1}$, respectively (where n depends on the context). Generalized inequalities such as $\boldsymbol{\alpha} \geq \mathbf{0}$ are understood coordinate-wise, i.e., $\alpha_i \geq 0$ for all i. In the whole thesis, it is understood that $\frac{x}{0} = 0$ if $x = 0$ and ∞ elsewise. We also employ the convention $\frac{\infty}{\infty} := 1$. Finally, for $p \in [1, \infty]$ we use the standard notation p^* to denote the conjugate of p, that is, $p^* \in [1, \infty]$ and $\frac{1}{p} + \frac{1}{p^*} = 1$, and $\mathbb{R}_+$ denotes nonnegative reals. Indicator functions are denoted by $\mathbb{1}$. We denote the set of nonnegative reals by $\mathbb{R}_+$.

2 A Unifying View of Multiple Kernel Learning

In this chapter, we cast multiple kernel learning into a unified framework. We show that it comprises many popular MKL variants currently discussed in the literature, including seemingly different ones. Our approach is based on regularized risk minimization (Vapnik, 1998). We derive generalized dual optimization problems without making specific assumptions regarding the norm 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. This chapter is based on mathematical preliminaries introduced in Appendix A.

The main **contributions** in this chapter are the following:

- we present a novel, unifying view of MKL, subsuming prevalent MKL approaches under a common umbrella
- this allows us to analyze the existing approaches jointly and is exemplified by deriving a unifying dual representation
- we show how prevalent models are contained in the framework as special cases.

Parts of this chapter are based on:

M. Kloft, U. Rückert, and P. L. Bartlett. A unifying view of multiple kernel learning. In *Proceedings of the European Conference on Machine Learning (ECML)*, 2010.

M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. Lp-norm multiple kernel learning. *Journal of Machine Learning Research (JMLR)*, 12:953–997, 2011.

2.1 A Regularized Risk Minimization Approach

We begin with reviewing the classical supervised learning setup, where we are given a labeled sample $\mathcal{D} = \{(x_i, y_i)\}_{i=1...,n}$ with x_i lying in some input space $\mathcal{X}$ and y_i in some output space $\mathcal{Y} \subset \mathbb{R}$. The goal in supervised learning is to find a hypothesis $f \in H$ that has a low error rate on new and unseen data. An often-employed approach to this problem is *regularized risk minimization* (RRM), where a minimizer

$$f^* \in \operatorname{argmin}_f \Omega(f) + CL_n(f)$$

is found. Hereby $L_n(f) = \sum_{i=1}^n l\left(f(x_i), y_i\right)$ is the (cumulative) *empirical loss* of a hypothesis f with respect to a function $l : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$ (called *loss function*); $\Omega : H \to \mathbb{R}$ is a mapping (called *regularizer*), and λ is a positive parameter. The name *risk* minimization stems from the fact that L_n is n-times the empirical risk. We can interpret RRM as minimizing a trade-off between the empirical loss (to classify the training data well) and a regularizer (to penalize the complexity of f and thus avoid overfitting), where the trade-off is controlled by λ.

Single-kernel approaches to RRM consider linear models of the form

$$f_{\boldsymbol{w},b}(x) = \langle \boldsymbol{w}, \phi(x) \rangle + b$$

together with a (possibly non-linear) mapping $\phi : \mathcal{X} \to \mathcal{H}$ to a Hilbert space $\mathcal{H}$ and regularizers

$$\Omega(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2 \tag{2.1}$$

(denoting by $\|\boldsymbol{w}\|_2$ the Hilbert-Schmidt norm in $\mathcal{H}$), which allows to "kernelize" (Schölkopf et al., 1998) the resulting models and algorithms, that is, formulating them solely in terms of inner products $k(x, x') := \langle \phi(x), \phi(x') \rangle$ in $\mathcal{H}$.

In *multiple kernel learning*, the feature mapping ϕ decomposes into M different feature mappings $\phi_m : \mathcal{X} \to \mathcal{H}_m,\ m = 1, \dots M$:

$$\phi : \begin{array}{ccc} \mathcal{X} & \to & \mathcal{H} \\ x & \mapsto & \big(\phi_1(x), \dots, \phi_M(x)\big) \end{array} .$$

Thereby, each ϕ_m gives rise to a kernel k_m so that the particular multiple kernels k_m are connected with the "joint" kernel k (the one corresponding to the composite feature map ϕ) by the simple equation

$$k = \sum_{m=1}^{M} k_m \ . \tag{2.2}$$

As with every decomposition one can argue that nothing is won by writing the feature map and the kernel as above. Indeed, in order to exploit the additional structure we can extend the regularizer (2.1) to

$$\Omega_{\text{MKL}}(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_{2,O}^2$$

where $\|\cdot\|_{2,O}$ denotes the $2, O$ block-norm is defined by

$$\|\boldsymbol{w}\|_{2,O} := \Big\| \Big(\|\boldsymbol{w}_1\|_2 , \dots, \|\boldsymbol{w}_M\|_2 \Big) \Big\|_O$$

and $\|\cdot\|_O$ is an arbitrary norm on $\mathbb{R}^M$. This allows to kernelize the resulting models and algorithms in terms of the multiple kernels k_m instead of the joint kernel k as defined in (2.2). The general multiple kernel learning RRM problem thus becomes

Problem 2.1 (Primal MKL Problem)**.**

$$\begin{array}{ll} \inf\limits_{\boldsymbol{w},b,\boldsymbol{t}} & \frac{1}{2}\,\|\boldsymbol{w}\|_{2,O}^2 + C \sum\limits_{i=1}^{n} l\big(t_i,\ y_i\big) \\ \text{s.t.} & \forall i: \ \ \langle \boldsymbol{w}, \phi(x_i) \rangle + b = t_i \ . \end{array} \tag{P}$$

The above primal generalizes multiple kernel learning to arbitrary convex loss functions and norms. Note that if the loss function is continuous (e.g., hinge loss) and the regularizer is such that its level sets form compact sets (e.g. $\ell_{2,p}$-norm, $p \geq 1$), then the supremum is in fact a maximum (this can be seen by rewriting the objective as a constrained optimization problem).

2.2 Dual Problem

In this section, we study the generalized MKL approach of the previous section in the dual space. Dual optimization problems deliver insight into the nature of an optimization problem. For example, they allow for computing the duality gap, which can be used as a stopping criterion at optimization time or to evaluate the quality of the numerical solutions by retrospect. Also, optimization can sometimes be considerably easier in the dual space.

We start the derivation by introducing Lagrangian multipliers $\boldsymbol{\alpha} \in \mathbb{R}^n$. We assume that the loss function is convex, so that (P) is a *convex optimization problem* (Boyd and Vandenberghe, 2004), and thus by the strong Lagrangian duality principle the optimal value of the primal (P) equals the one of the associated Lagrangian saddle point problem (note that the given constraints are only linear *equality* ones, so that constraint qualification trivially holds),

$$\sup_{\boldsymbol{\alpha}} \quad \inf_{\boldsymbol{w},b,\boldsymbol{t}} \quad \mathcal{L}\big(\boldsymbol{\alpha}, (\boldsymbol{w}, b, \boldsymbol{t})\big) \tag{2.3}$$

with Lagrangian function

$$\mathcal{L}\big(\boldsymbol{\alpha}, (\boldsymbol{w}, b, \boldsymbol{t})\big) \;=\; \frac{1}{2}\,\|\boldsymbol{w}\|_{2,O}^2 + C\sum_{i=1}^n l\big(t_i,\; y_i\big) + \sum_{i=1}^n \alpha_i\big(t_i - \langle \boldsymbol{w}, \phi(x_i)\rangle - b\big)\;.$$

A standard approach in convex optimization and machine learning is to invoke the KKT conditions to remove the dependancy on the primal variables in the above problem. But because the objective is not differentiable for general norms and losses (except for b where $\nabla_b \mathcal{L} = 0$ leads to $\mathbf{1}^\top \boldsymbol{\alpha} = 0$), we follow a different approach based on conjugate functions.[4] We start by rewriting (2.3) as

$$\sup_{\boldsymbol{\alpha}:\,\mathbf{1}^\top\boldsymbol{\alpha}=0} -C\sum_{i=1}^n \sup_{t_i}\Big(-\frac{\alpha_i t_i}{C} - l\big(t_i, y_i\big)\Big) - \sup_{\boldsymbol{w}}\Big(\Big\langle \boldsymbol{w}, \sum_{i=1}^n \alpha_i\phi(x_i)\Big\rangle - \frac{1}{2}\,\|\boldsymbol{w}\|_{2,O}^2\Big).$$

The Fenchel-Legendre conjugate function of a function $g : \mathbb{R}^k \to \mathbb{R}$ is defined by $g^* : \boldsymbol{t} \mapsto \sup_{\boldsymbol{u}} \boldsymbol{u}^\top\boldsymbol{t} - g(\boldsymbol{u})$. If g is a loss function then we consider the conjugate in the first argument and call it *dual loss*. Equipped with this notation, we can rewrite the above problem as

$$\sup_{\boldsymbol{\alpha}:\,\mathbf{1}^\top\boldsymbol{\alpha}=0} -C\sum_{i=1}^n l^*\Big(-\frac{\alpha_i}{C}, y_i\Big) - \left(\frac{1}{2}\Big\|\sum_{i=1}^n \alpha_i\phi(x_i)\Big\|_{2,O}^2\right)^*.$$

We note that for any norm it holds $\big(\frac{1}{2}\|\cdot\|^2\big)^* = \frac{1}{2}\|\cdot\|_*^2$ where

$$\frac{1}{2}\|\cdot\|_* :\;\; \boldsymbol{w} \mapsto \sup_{\boldsymbol{u}:\|\boldsymbol{u}\|\le 1} \boldsymbol{u}^\top\boldsymbol{w} \tag{2.4}$$

[4] Although we developed the approach for the purpose of MKL dualization, it might also be useful outside the scope of MKL; for example, the RRM dualization approach given in Rifkin and Lippert (2007) is contained in our framework as a special case but in contrast to them we can employ arbitrary (i.e., not necessarily strictly) positive-semidefinite kernels.

denotes the dual norm (Boyd and Vandenberghe, 2004, , 3.26–3.27). Denoting the dual norm of $\|\cdot\|_O$ by $\|\cdot\|_{O^*}$, we can remark that the norm dual to $\|\cdot\|_{2,O}$ is $\|\cdot\|_{2,O^*}$ (e.g., Aflalo et al., 2011). Thus we can further rewrite the above, resulting in the following *dual* MKL optimization problem which now solely depends on $\boldsymbol{\alpha}$:

Problem 2.2 (Dual MKL Problem)**.**

$$\sup_{\boldsymbol{\alpha}:\, \mathbf{1}^\top \boldsymbol{\alpha}=0} -C \sum_{i=1}^{n} l^*\Big(-\frac{\alpha_i}{C}, y_i\Big) - \frac{1}{2}\Big\| \sum_{i=1}^{n} \alpha_i \phi(x_i)\Big\|^2_{2,O^*}. \tag{D}$$

The above dual generalizes multiple kernel learning to arbitrary convex loss functions and arbitrary block-structured norms where the inner norm is a Hilbert-Schmidt norm.

Discussion We note that (like in the primal) we have a decomposition of the above dual into a loss term (on the left hand side) and a block-structured regularization term (right hand side). A difference to the primal is that the decomposition is in terms of the *dual* loss/regularizer. An advantage of the compact representation of the above dual is that for a specific loss/regularizer pair $(l, \|\cdot\|)$ we just have to plug the dual loss / dual regularizer in (D) in order to obtain the dual MKL optimization problem. We illustrate this by some examples of loss functions and regularizers in the next section.

2.3 Recovering Prevalent MKL Formulations as Special Cases

In this section, we show that existing MKL-based learners are subsumed by the generalized formulations in (P) and compute their dual representations (D). To do so, we need to first compute the dual losses and dual regularizers. To this aim, we present a table of loss functions and their duals (see Table 2.1). The table can be verified by elementary calculations from the definition of the conjugate.

2.3.1 Support Vector Machines with Unweighted-Sum Kernels

Clearly, by considering the hinge loss $l(t, y) = \max(0, 1 - ty)$ and the regularizer $\|\boldsymbol{w}\|_{2,2}$, the support vector machine using a uniform combination of kernels is a special case of our generalized formulation. It is instructive to compute the dual (D) for this simple example.

To this aim, we first note that the dual loss of the hinge loss is $l^*(t, y) = \frac{t}{y}$ if $-1 \leq \frac{t}{y} \leq 0$ and ∞ elsewise (see Table 2.1). Hence, for each i the term $l^*\left(-\frac{\alpha_i}{C},\ y_i\right)$ of the generalized dual (D) translates to $-\frac{\alpha_i}{C y_i}$ provided that $0 \leq \frac{\alpha_i}{y_i} \leq C$. We can now employ a variable substitution $\alpha_i^{\text{new}} = \frac{\alpha_i}{y_i}$ so that (D) reads

$$\max_{\boldsymbol{\alpha}} \quad \mathbf{1}^\top \boldsymbol{\alpha} - \frac{1}{2}\Big\| \sum_{i=1}^{n} \alpha_i y_i \phi(x_i)\Big\|^2_{2,2}, \quad \text{s.t.} \quad \boldsymbol{y}^\top \boldsymbol{\alpha} = 0 \ \text{ and } \ \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}, \tag{2.5}$$

Note that the regularizer $\|\boldsymbol{w}\|_{2,2}$ is just $\|\boldsymbol{w}\|_2$. Hence, the right-hand side of the last equation can be simply written as $\sum_{m=1}^{M} \boldsymbol{\alpha}^\top Y K_m Y \boldsymbol{\alpha}$, where $Y = \operatorname{diag}(\boldsymbol{y})$, and thus

Table 2.1: Loss functions and regularizers used in this thesis and corresponding conjugate functions.

	loss/regularizer $g(t,y)$	conjugate $g^*(t,y)$	used in[5]
hinge loss	$\max(0, 1-ty)$	$\frac{t}{y}$ if $-1 \leq \frac{t}{y} \leq 0$ and ∞ elsewise	[1]
squared loss	$\frac{1}{2}(y-t)^2$	$\frac{1}{2}t^2 + ty$	[2],[3]
unsigned hinge	$\max(0, 1-t)$	t if $-1 \leq t \leq 0$ and ∞ elsewise	[4]
ℓ_1-norm	$\frac{1}{2}\big(\sum_{m=1}^M \|\boldsymbol{w}_m\|_2\big)^2$	$\frac{1}{2}\big(\max_{m\in\{1,\ldots,\infty\}} \|\boldsymbol{w}\|_2\big)^2$	[5]
ℓ_p-norm	$\frac{1}{2}\|\boldsymbol{w}\|_p^2$	$\frac{1}{2}\|\boldsymbol{w}\|_{p^*}^2$	[6]
$\ell_{2,p}$ block-norm	$\frac{1}{2}\|\boldsymbol{w}\|_{2,p}^2$	$\frac{1}{2}\|\boldsymbol{w}\|_{2,p^*}^2$	[5],[7],[8][6]

we obtain from (2.5):

$$\begin{aligned} \sup_{\boldsymbol{\alpha}} \quad & \mathbf{1}^\top\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^\top YKY\boldsymbol{\alpha} \\ \text{s.t.} \quad & \boldsymbol{y}^\top\boldsymbol{\alpha} = 0, \quad \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1} \ . \end{aligned}$$

This is the usual dual SVM optimization problem using a uniform kernel combination $K = \sum_{m=1}^M K_m$ (Müller et al., 2001).

2.3.2 The Classical Quadratically Constrained Quadratic Program (QCQP)

A classical approach to multiple kernel learning, going back to the work of Lanckriet et al. (2004a) and Bach et al. (2004), is to employ regularizers of the form

$$\boldsymbol{w} \mapsto \frac{1}{2}\|\boldsymbol{w}\|_{2,1}^2 \tag{2.6}$$

to promote (blockwise) sparse solutions (many $\boldsymbol{w}_m$ are zero). Since $\boldsymbol{w}_m = 0$ means that the corresponding kernel is "switched off" and does not contribute to the decision function, the so-obtained solutions are *interpretable*.

We can view the classical sparse MKL as a special case of our unifying framework; to see this, note that the norm dual to ℓ_1 is ℓ_∞ (see Table 2.1). This means the right hand side of (D) translates to $\max_{m\in\{1,\ldots,M\}} \boldsymbol{\alpha}^\top YK_mY\boldsymbol{\alpha}$; subsequently, the maximum

[5] [1] Lanckriet et al. (2004a); [2] Yuan and Lin (2006); [3] Bach (2008b); [4] Sonnenburg et al. (2006a);
[5] Bach et al. (2004); [6] Kloft et al. (2009a); [7] Kloft et al. (2011); [8] Aflalo et al. (2011)

[6] Only for $p = 1$ in [5]; only for $1 < p < 2$ in [7]; only for $p > 2$ in [8]

can be expanded into a slack variable ξ, resulting in

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

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

2.3.3 A Smooth Variant of Group Lasso

Yuan and Lin (2006) studied the following regularized risk minimization problem known as the *group lasso*,

$$\min_{\boldsymbol{w}} \quad \frac{C}{2}\sum_{i=1}^{n}\left(y_i - \sum_{m=1}^{M}\left\langle \boldsymbol{w}_m, \phi_m(x_i)\right\rangle\right)^2 + \frac{1}{2}\sum_{m=1}^{M}\|\boldsymbol{w}_m\|_2 \tag{2.7}$$

for $\mathcal{H}_m = \mathbb{R}^{d_m}$. The above problem has been solved by active set methods in the primal (Roth and Fischer, 2008). We sketch an alternative approach based on dual optimization. First, we note that the dual of $l(t,y) = \frac{1}{2}(y-t)^2$ is $l^*(t,y) = \frac{1}{2}t^2 + ty$ and thus the corresponding group lasso dual according to (D) can be written as

$$\max_{\boldsymbol{\alpha}} \quad \boldsymbol{y}^\top \boldsymbol{\alpha} - \frac{1}{2C}\|\boldsymbol{\alpha}\|_2^2 - \frac{1}{2}\left\|\left(\boldsymbol{\alpha}^\top Y K_m Y \boldsymbol{\alpha}\right)_{m=1}^{M}\right\|_\infty, \tag{2.8}$$

which can be expanded into the following QCQP

$$\begin{aligned} \sup_{\boldsymbol{\alpha},\xi} \quad & \boldsymbol{y}^\top \boldsymbol{\alpha} - \frac{1}{2C}\|\boldsymbol{\alpha}\|_2^2 - \xi \\ \text{s.t.} \quad & \forall m: \ \frac{1}{2}\boldsymbol{\alpha}^\top Y K_m Y \boldsymbol{\alpha} \le \xi. \end{aligned} \tag{2.9}$$

For small n, the latter formulation can be handled rather efficiently by QCQP solvers. However, the many quadratic constraints, caused by the non-smooth ℓ_∞-norm in the objective, still are computationally demanding. As a remedy, we propose the following unconstrained variant based on ℓ_p-norms ($1 < p < 2$), given by

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

Because of the smoothness of the $\ell_{p>1}$-norm, the above objective function is differentiable in any $\boldsymbol{\alpha} \in \mathbb{R}^n$. Thus the above optimization problem can be solved very efficiently by Newton descent methods (Nocedal and Wright, 2006) such as the limited memory quasi-Newton method of (Liu and Nocedal, 1989).

2.3.4 Density Level-Set Estimation

Density level-set estimators such as the one-class support vector machine (Schölkopf et al., 2001) are frequently used for anomaly, novelty, and outlier detection tasks (see, for example, Markou and Singh, 2003a,b). The one-class SVM can be cast into our multi-kernel framework by employing loss functions of the form $l(t) = \max(0, 1-t)$:

$$\begin{aligned} \inf_{\boldsymbol{w},b,\boldsymbol{t}} \quad & \frac{1}{2}\left\|\boldsymbol{w}\right\|_{2,O}^2 + C\sum_{i=1}^n \max(0, 1-t_i) \\ \text{s.t.} \quad & \forall i: \ \ \langle \boldsymbol{w}, \phi(x_i)\rangle + b = t_i \ . \end{aligned}$$

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

$$\sup_{\boldsymbol{\alpha}:\boldsymbol{0}\leq\boldsymbol{\alpha}\leq C\boldsymbol{1}} \quad \boldsymbol{1}^\top\boldsymbol{\alpha} - \frac{1}{2}\left\|\sum_{i=1}^n \alpha_i\phi(x_i)\right\|_{2,O^*}^2 .$$

This is studied in Sonnenburg et al. (2006a) for $\ell_{2,1}$-norms and in Kloft et al. (2009b) for $\ell_{2,p}$-norms ($1 \leq p \leq 2$).

2.3.5 Hierarchical Kernel Learning

Often multiple kernels come together with a block structure; this frequently happens, for example, when multiple types of the kernels are given and each type is realized for a number of parameter values (see the bioinformatics application in Section 5.4.1 and the computer vision application in Section 5.5 as examples). Hierarchical kernel learning was recently studied using two levels of hierarchies: let $I_1, \ldots, I_g \subset \{1, \ldots, M\}$ be pairwise disjoint index sets; then the following penalty was studied

$$\Omega(\boldsymbol{w}) := \frac{1}{2}\left\|\Big(\left\|\left(\|\boldsymbol{w}_i\|_2\right)_{i\in I_1}\right\|_q, \ldots, \left\|\left(\|\boldsymbol{w}_i\|_2\right)_{i\in I_g}\right\|_q \Big)\right\|_p^2 .$$

This was considered in Szafranski et al. (2010) for $p, q \in [1, 2]$ and in Aflalo et al. (2011) for $p =\in [2, \infty]$ and $q = 1$. Hierarchical kernel learning is contained in our unifying framework: we obtain a dual representation by noting that the conjugate regularizer is

$$\Omega(\boldsymbol{w})^* = \frac{1}{2}\left\|\Big(\left\|\left(\|\boldsymbol{w}_i\|_2\right)_{i\in I_1}\right\|_{q^*}, \ldots, \left\|\left(\|\boldsymbol{w}_i\|_2\right)_{i\in I_g}\right\|_{q^*} \Big)\right\|_{p^*}^2 .$$

so that the general dual (D) translates into the following dual problem:

$$\sup_{\boldsymbol{\alpha}:\,\boldsymbol{1}^\top\boldsymbol{\alpha}=0} -C\sum_{i=1}^n l^*\Big(-\frac{\alpha_i}{C}, y_i\Big) - \frac{1}{2}\left\|\left(\left\|\left(\left\|\sum_{i=1}^n \alpha_i\phi_m(x_i)\right\|_2\right)_{m\in I_j}\right\|_{q^*}\right)_{j=1}^g\right\|_{p^*} .$$

We can remark here that, from our unifying point of view, there is no need for the distinction of cases according to the ranges of p and q as imposed in Szafranski et al. (2010) and Aflalo et al. (2011): the whole range $p, q \in [1, \infty]$ can be analyzed simultaneously in our framework.

2.3.6 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 provide estimates of the interactions of the kernel matrices $K_1, \ldots, K_M$ in the form of an $M \times M$ matrix E. For example, if the kernels are related by an underlying graph structure, E could be the graph Laplacian (von Luxburg, 2007) encoding the similarities of the kernel matrices.

Alternatively, E could be estimated from data by computing the pairwise kernel alignments $E_{ij} = \frac{<K_i, K_j>}{\|K_i\| \, \|K_j\|}$ (given an inner product on the space of kernel matrices such as the Frobenius dot product).

In a third scenario, E could be a diagonal matrix encoding the a priori importance of kernels—it might be known from pilot studies that a subset of the employed kernels is inferior to the remaining ones.

All those scenarios are subsumed by the proposed framework by considering non-isotropic regularizers of the form

$$\Omega(\boldsymbol{w}) := \frac{1}{2}\|\boldsymbol{w}\|_{E^{-1}}^2$$

where

$$\|\boldsymbol{w}\|_{E^{-1}} := \sqrt{\boldsymbol{w}^\top E^{-1} \boldsymbol{w}}$$

for some $E \succ 0$, where E^{-1} is the matrix inverse of E. The dual norm of $\|\cdot\|_{E^{-1}}$ is $\|\cdot\|_E$ (this is easily verified by setting the gradient of the conjugate of $\frac{1}{2}\|\cdot\|_{E^{-1}}^2$ to zero) so that we obtain from (D) the dual optimization problem

$$\sup_{\boldsymbol{\alpha}:\mathbf{1}^\top\boldsymbol{\alpha}=\mathbf{0}} \; - C \sum_{i=1}^{n} l^* \left(-\frac{\alpha_i}{C},\, y_i\right) - \frac{1}{2}\left\| \sum_{i=1}^{n} \alpha_i \phi(x_i) \right\|_E^2,$$

which is a non-isotropic MKL problem. The usage of non-isotropic MKL was first proposed in Varma and Ray (2007) for the very simple case of E being a diagonal matrix and generalized in Kloft et al. (2011).

2.4 Summary and Discussion

The standard view of multiple kernel learning (introduced in the introduction) has a few limitations: first, obviously, it is incomplete since it does not account for $p > 2$ (Kloft et al., 2010a); second, there exist convex MKL variants (again, corresponding to $p > 2$) that, in the standard view, cannot be represented in convex form, although inherently being convex (Aflalo et al., 2011); third, as it will turn out, the standard view is inconvenient for a theoretical generalization analysis such as the one carried out in Chapter 4 of this thesis.

As a remedy, we developed a rigorous mathematical framework for the problem of multiple kernel learning in this chapter. It comprises most existing lines of research in that area, including very recent and seemingly different ones (e.g., the one considered in Aflalo et al., 2011). Deviating from standard introductions, we phrased MKL as

a general optimization criterion based on structured regularization, covering and also unifying existing formulations under a common umbrella. By plugging arbitrary convex loss functions and norms into the general framework, many existing approaches can be recovered as instantiations of our model.

The unifying framework allows us to analyze a large variety of MKL methods jointly, as exemplified by deriving a general dual representation of the criterion, without making assumptions on the employed norms and losses, besides the latter being convex. This delivers insights into connections between existing MKL formulations and, even more importantly, can be used to derive *novel* MKL formulations as special cases of our framework, as done in the next part of the thesis, where we propose ℓ_p-norm multiple kernel learning. We note that in the most basic special case, the classical ℓ_1-norm MKL formulation of Lanckriet et al. (2004a) is recovered by plugging the hinge loss and the ℓ_1-norm into the framework. Historically, the structured view of classical MKL is known since Bach et al. (2004). Here, we show that, more generally, the whole family of MKL methods can be viewed as *structured regularization* (Obozinski et al., 2011), of which we argue that it is a more elegant way to view MKL than the standard approach.

Part II

ℓ_p-norm Multiple Kernel Learning

II ℓ_p-norm Multiple Kernel Learning

In the previous chapter we presented a general view of MKL. However, at some point we have to make a particular choice of a norm, for example, in order to employ MKL in practical applications.

The main contribution of this thesis is a novel MKL formulation called *ℓ_p-norm multiple kernel learning*. Our interest in this stems from the fact that (in contrast to the prevalent MKL variants) it has a good chance to improve on the trivial uniform-kernel-combination baseline in practical applications (we show this later in the Chapter 5 of this thesis).

We obtain ℓ_p-norm MKL from the unifying MKL framework by using the regularizer

$$\Omega_{\text{MKL}}(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_{2,p}^2, \qquad p \in \{1, \ldots, \infty\},$$

where the *$\ell_{2,p}$-norm* is defined as

$$\|\boldsymbol{w}\|_{2,p} := \left(\sum_{m=1}^{M} \|\boldsymbol{w}_m\|_2^p \right)^{\frac{1}{p}}$$

for $p \in [1, \infty[$ and $\|\boldsymbol{w}\|_{2,p} := \sup_{m=1,\ldots,M} \|\boldsymbol{w}_m\|_2$ for $p = \infty$. Plugging this into the unifying optimization problems (Problems 2.1 and 2.2) we obtain the ℓ_p-norm MKL primal and dual problems:

Problem II.3 (ℓ_p-NORM MKL)**.** *For any $p \in [1, \infty]$,*

$$\begin{aligned} \inf_{\boldsymbol{w},b,\boldsymbol{t}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_{2,p}^2 + C\sum_{i=1}^{n} l\big(t_i,\ y_i\big) && \text{(PRIMAL)} \\ \text{s.t.} \quad & \forall i: \ \langle \boldsymbol{w}, \phi(x_i)\rangle + b = t_i \ , \end{aligned}$$

$$\sup_{\boldsymbol{\alpha}:\, \mathbf{1}^\top\boldsymbol{\alpha}=0} \ - C\sum_{i=1}^{n} l^*\Big(-\frac{\alpha_i}{C}, y_i\Big) - \frac{1}{2}\Big\| \sum_{i=1}^{n} \alpha_i \phi(x_i) \Big\|_{2,p^*}^2 . \qquad \text{(DUAL)}$$

The key difference of $\ell_{p>1}$-norm MKL to previous MKL approaches, which are based on ℓ_1-norms, is that the obtained weight vectors are unlikely to be *sparse*. This was already indicated in the introduction and is now discussed in more detail here: the optimal solution of the ℓ_p-norm MKL optimization problem is attained when the norm

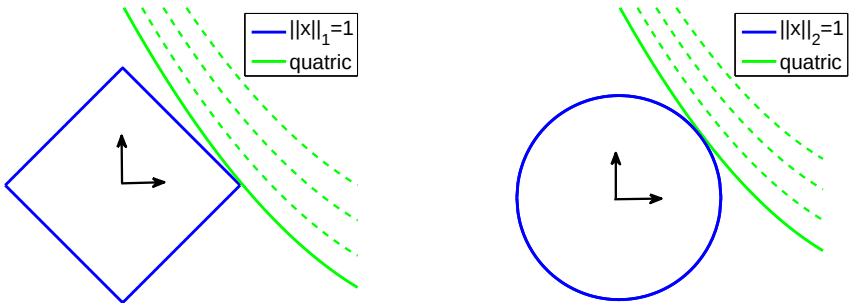


Figure 2.1: Comparison of ℓ_1- (LEFT) and ℓ_2- (RIGHT) regularized problems. The blue and green lines show level sets of the ℓ_1-norm and a quadratic function, respectively.

constraint and the level sets of loss function touch each other (see Figure 2.1). If the loss function is convex (illustrated here for a quadratic loss function) and the norm is a $\ell_{2,1}$-norm, the point of intersection is likely to be at one of the corners of the square (shown on the figure to the left). Because the corners are likely to have zero-entries, sparsity is to be expected. This is in contrast to $\ell_{2,p}$-norm regularization with $p > 1$, where, in general, a non-sparse solution is to be expected (shown on the figure to the right).

Why ℓ_p-norms?

A naturally arising question is why (out of the set of all norms) focusing on an ℓ_p-norm? A reason for this is that, without any prior knowledge, an ℓ_p-norm is a natural choice—in contrast, for example, non-isotropic norms rely on prior knowledge of the relative importance of the particular kernels or the interactions *between* the kernels. Also, our optimization algorithms and the theoretical analysis presented later in this thesis make use of particular properties of the ℓ_p-norm that are not valid for any arbitrary norm.

Alternative Formulation

In the previous chapter, we formulated MKL as a block-norm-regularized risk minimization problem. Here, we present an alternative but equivalent view of ℓ_p-norm MKL as learning an "optimal" (linear) kernel combination

$$K_{\boldsymbol{\theta}} = \sum_{m=1} \theta_m K_m$$

from a candidate set of kernels $\{K_1, \ldots, K_M\}$ subject to $\|\boldsymbol{\theta}\|_p = 1$. We now show that this formulation naturally arises from the previously considered formulation of Problem II.3. We need to treat the cases $p \in [1, 2]$ and $p \in [2, \infty]$ separately to ensure that the occurring "norms" indeed are norms.

The case $p \in [1, 2[$

We first deal with the case $p \in [1, 2[$, which corresponds to the established standard view of MKL. To this aim, we reconsider the regularizer of Problem II.3 and note that for $p \in [1, 2]$ we can rewrite it as

$$\begin{aligned} \Big\| \sum_{i=1}^{n} \alpha_i \phi(x_i) \Big\|_{2,p^*}^2 &= \Big\| \big(\boldsymbol{\alpha}^\top K_m \boldsymbol{\alpha} \big)_{m=1}^{M} \Big\|_{p^*/2} \\ &\overset{(2.4)}{=} \sup_{\boldsymbol{\theta}:\, \|\boldsymbol{\theta}\|_{(p^*/2)^*} \leq 1} \; \sum_{m=1}^{M} \theta_m \boldsymbol{\alpha}^\top K_m \boldsymbol{\alpha} \end{aligned} \tag{II.1}$$

where we use the definition of the dual norm (Equation (2.4)). Note that the choice of $p \in [1, 2]$ ensures that the above "norm" really is a norm: indeed, $p \in [1, 2]$ implies that $p^*/2$ is in the valid interval $[1, \infty]$. We also note that $(p^*/2)^* = \frac{p}{2-p}$ and that the optimal $\boldsymbol{\theta}$ is nonnegative so that we can use (II.1) to rewrite Problem II.3 as

$$\inf_{\boldsymbol{\theta}:\, \|\boldsymbol{\theta}\|_{p/(2-p)} \leq 1} \; \textsc{svm}(K_{\boldsymbol{\theta}}), \qquad \text{s.t.} \quad K_{\boldsymbol{\theta}} = \sum_{m=1}^{M} \theta_m K_m \;, \tag{II.2}$$

where we use the shorthand

$$\textsc{svm}(K_{\boldsymbol{\theta}}) \;:=\; \sup_{\boldsymbol{\alpha}:\, \mathbf{1}^\top \boldsymbol{\alpha} = 0} \; -C \sum_{i=1}^{n} l^*\Big(-\frac{\alpha_i}{C}, y_i \Big) - \frac{1}{2} \boldsymbol{\alpha}^\top K_{\boldsymbol{\theta}} \boldsymbol{\alpha} \tag{II.3}$$

to denote the optimal value of the SVM optimization problem. Note that we exchanged the sequence of minimization and maximization to obtain (II.2), which is justified by Sion's Minimax theorem (Sion, 1958). Equation (II.2) gives an alternative formulation of the ℓ_p-norm MKL problem: training a support vector machine which simultaneously also optimizes over the optimal kernel combination (subject to a norm constraint on the combination coefficients to avoid overfitting).

The case $p \in]2, \infty]$

Unfortunately, we cannot use the above approach in the case $p \in [2, \infty]$ as then the norm parameter $p/(2-p)$ lies outside the valid range of $[1, \infty]$. However, we can use a similar argument as above by considering the *primal* of Problem II.3 instead of the dual: the primal regularizer of Problem II.3 can be rewritten as

$$\begin{aligned} \|\boldsymbol{w}\|_{2,p}^2 &= \Big\| \Big(\|\boldsymbol{w}_m\|_2^2 \Big)_{m=1}^{M} \Big\|_{p/2} \overset{(2.4)}{=} \sup_{\boldsymbol{\theta}:\, \|\boldsymbol{\theta}\|_{(p/2)^*} \leq 1} \; \sum_{m=1}^{M} \theta_m \|\boldsymbol{w}_m\|_2^2 \\ &= \sup_{\boldsymbol{\theta}:\, \sum_m \theta_m^{-(p/2)^*} \leq 1} \; \sum_{m=1}^{M} \theta_m^{-1} \|\boldsymbol{w}_m\|_2^2 \;. \end{aligned} \tag{II.4}$$

since $p/2$ is in the valid interval $[1,\infty]$ for $p \in [2,\infty]$. We note that $-(p/2)^* = p/(2-p)$ and that the optimal $\boldsymbol{\theta}$ is nonnegative; hence, the primal of Problem II.3 translates into

$$\sup_{\boldsymbol{\theta}:\, \sum_m \theta_m^{p/(2-p)} \leq 1} \quad \text{SVM}(K_{\boldsymbol{\theta}}), \qquad \text{s.t.} \quad K_{\boldsymbol{\theta}} = \sum_{m=1}^{M} \theta_m K_m \ ,$$

$$\text{with} \qquad \text{SVM}(K_{\boldsymbol{\theta}}) \ := \ \inf_{\boldsymbol{w},b} \quad \frac{1}{2}\sum_{m=1}^{M} \theta_m^{-1} \left\|\boldsymbol{w}_m\right\|_2^2 + C\sum_{i=1}^{n} l\big(\langle \boldsymbol{w}, \phi(x_i)\rangle + b,\ y_i\big) \ .$$

Remark

Note that in the above definition of the SVM function there is no collision with the definition in (II.3) since both formulations are dual to each other, for any fixed $\boldsymbol{\theta} \geq \mathbf{0}$. One way to see this is by introducing slack variables $t_i := \langle \boldsymbol{w}, \phi(x_i)\rangle + b$ to write the above as

$$\text{SVM}(K_{\boldsymbol{\theta}}) \ = \ \inf_{\boldsymbol{w},b,\boldsymbol{t}} \quad \frac{1}{2}\sum_{m=1}^{M} \theta_m^{-1} \left\|\boldsymbol{w}_m\right\|_2^2 + C\sum_{i=1}^{n} l\big(t_i,\ y_i\big)$$

and then incorporating the constraints by Lagrangian multipliers α_i:

$$\sup_{\boldsymbol{\alpha}} \quad \inf_{\boldsymbol{w},b,\boldsymbol{t}} \quad \frac{1}{2}\sum_{m=1}^{M} \theta_m^{-1} \left\|\boldsymbol{w}_m\right\|_2^2 + C\sum_{i=1}^{n} l\big(t_i,\ y_i\big) + \sum_{i=1}^{n} \alpha_i\big(t_i - \langle \boldsymbol{w}, \phi(x_i)\rangle - b\big) \ .$$

The KKT conditions, which can be used to compute the decision function and to recover the treshold b, hold for the pair $(\boldsymbol{w}, \boldsymbol{\alpha})$ and yield

$$\forall\, m = 1, \ldots, M: \quad \left\|\boldsymbol{w}_m\right\|_2^2 = \theta_m^2 \boldsymbol{\alpha} K_m \boldsymbol{\alpha} \ . \tag{II.5}$$

Note, furthermore, that applying the KKT conditions to (2.3) for ℓ_p-norms yields:

$$\exists\, c > 0: \quad \left\|\boldsymbol{w}_m\right\| = c\,(\boldsymbol{\alpha} K_m \boldsymbol{\alpha})^{\frac{1}{2(p-1)}}, \tag{II.6}$$

The remainder of the dualization is analogue to what we have seen above and yields (II.3).

Putting the Pieces Together...

Putting things together, we obtain the following alternative MKL optimization problem.

Problem II.4 (ℓ_p-NORM MKL, ALTERNATIVE FORMULATION)**.** *The alternative formulation of ℓ_p-norm multiple kernel learning is given by*

$$\inf_{\boldsymbol{\theta} \geq \mathbf{0}} \quad \text{SVM}(K_{\boldsymbol{\theta}}), \qquad (\text{ if } p \in [1,2[\ )$$

$$\sup_{\boldsymbol{\theta} \geq \mathbf{0}} \quad \text{SVM}(K_{\boldsymbol{\theta}}), \qquad (\text{ if } p \in]2,\infty]\ )$$

$$\text{s.t.} \quad K_{\boldsymbol{\theta}} = \sum_{m=1}^{M} \theta_m K_m, \qquad \sum_m \theta_m^{p/(2-p)} \leq 1,$$

where SVM *denotes the optimal objective value of the SVM optimization problem,*

$$\begin{aligned}\text{SVM}(K_{\boldsymbol{\theta}}) \quad &:= \quad \inf_{\boldsymbol{w},b} \qquad \frac{1}{2}\sum_{m=1}^{M}\theta_m^{-1}\left\|\boldsymbol{w}_m\right\|_2^2 + C\sum_{i=1}^{n} l\big(\langle \boldsymbol{w}, \phi(x_i)\rangle + b,\ y_i\big) \\ &= \sup_{\boldsymbol{\alpha}:\,\mathbf{1}^\top\boldsymbol{\alpha}=0} \qquad -C\sum_{i=1}^{n} l^*\Big(-\frac{\alpha_i}{C}, y_i\Big) - \frac{1}{2}\boldsymbol{\alpha}^\top K_{\boldsymbol{\theta}}\boldsymbol{\alpha}\ .\end{aligned}$$

The above problem can be interpreted as finding or "learning" an optimal kernel combination from a set of kernels where the quality of the mixtures is evaluated in terms of the SVM objective function. This alternative view is much in the original *learning-kernels* spirit of Lanckriet et al. (2004a): "Learning the kernel matrix with semi-definite programming".

Remaining Contents of this Part

In the remainder of this part we derive optimization algorithms for ℓ_p-norm MKL and apply them in order to empirically analyze the generalization performance of ℓ_p-norm MKL in controlled artificial environments as well as real-world applications from the domains of bioinformatics and computer vision. We also investigate ℓ_p-norm MKL theoretically and show that for a large range of learning scenarios it enjoys stronger generalization guarantees than classical MKL and the SVM using a uniform kernel combination.

3 Algorithms

In this chapter, we present two optimization algorithms for the ℓ_p-norm MKL problem II.3. Each algorithm has its own advantage: while the first one is provably convergent, easier to implement, and also to modify, due to its modular design, the second one is expected to be faster in practice and also less memory-intensive. For the sake of performance, both algorithms were implemented in C++ and made available as a part of the open source machine learning toolbox SHOGUN (Sonnenburg et al., 2010), which also contains interfaces to MATLAB, Octave, Phyton, and R.

Both algorithms are based on the alternative ℓ_p-norm MKL formulation (Problem II.4) in contrast to the original problem (Problem II.3). This is because the former consists of an "inner" and an "outer" optimization problem, where the inner problem is a standard SVM optimization problem. This has the advantage that existing (highly optimized) SVM solvers can be exploited.

We remark that the proposed algorithms can, in particular, be used to optimize the classical $\ell_{p=1}$-norm MKL formulation. In the computational experiments at the end of this chapter, we show that the proposed algorithms outperform the prevalent state-of-the-art solvers by up to two magnitudes.

The main **contributions** in this chapter are the following:

- We present a novel optimization method based on a simple, analytical formula, which, in particular, can be used to optimize classical MKL.
- We prove its convergence for $p > 1$.
- We implement the algorithm as an interleaved chunking optimizer in C++ within the SHOGUN toolbox with interfaces to MATLAB, Octave, Phyton, and R.
- We show that our implementation allows for training with up to ten thousands of data points and thousands of kernels, while the state-of-the-art approaches run already out of memory with a few thousand of data points and hundreds of kernels.
- Even for moderate sizes of the training and kernel sets, our approach outperforms prevalent ones by up to two magnitudes.

Parts of this chapter are based on:

M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Müller, and A. Zien. Efficient and accurate lp-norm multiple kernel learning. In *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, pages 997–1005. MIT Press, 2009.

M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. Lp-norm multiple kernel learning. *Journal of Machine Learning Research (JMLR)*, 12:953–997, 2011.

3.1 Block Coordinate Descent Algorithm

The main idea of the first approach is to divide the set of optimization variables (which is $\{\theta_1, \ldots, \theta_M, \alpha_1, \ldots, \alpha_n\}$) into two groups—the set $\{\theta_1, \ldots, \theta_M\}$ on one hand and the set $\{\alpha_1, ..., \alpha_M\}$ on the other—and then alternating the optimization with respect to $\boldsymbol{\theta}$ with the one with respect to $\boldsymbol{\alpha}$.

We observe that in the $\boldsymbol{\alpha}$-step this boils down to training a standard SVM. In contrast, the $\boldsymbol{\theta}$-step can simply be performed by means of an analytical formula, as the following proposition shows.

Proposition 3.1. *Given any (possibly suboptimal) $\boldsymbol{w} \neq \mathbf{0}$ in the objective of Problem II.4, the optimal $\boldsymbol{\theta}$ is attained for*

$$\forall\, m = 1, \ldots, M: \qquad \theta_m = \frac{\|\boldsymbol{w}_m\|_2^{2-p}}{\Big(\sum_{m'=1}^{M} \|\boldsymbol{w}_{m'}\|_2^p\Big)^{(2-p)/p}} \ . \qquad (\text{ if } p \in [1,2[\,)$$

Assume at least one of the kernel matrices K_m is strictly positive-definite. Then, given any (possibly suboptimal) $\boldsymbol{\alpha} \neq \mathbf{0}$, the optimal $\boldsymbol{\theta}$ is attained for

$$\forall\, m = 1, \ldots, M: \qquad \theta_m = \frac{\big(\boldsymbol{\alpha}^\top K_m \boldsymbol{\alpha}\big)^{-\frac{2-p}{2-2p}}}{\Big(\sum_{m'=1}^{M} \big(\boldsymbol{\alpha}^\top K_{m'} \boldsymbol{\alpha}\big)^{-\frac{p}{2-2p}}\Big)^{(2-p)/p}} \ . \qquad (\text{ if } p \in]2,\infty]\,)$$

Proof To start the proof, we consider Problem II.4, fix the variables $\boldsymbol{w}, b$, and only optimize w.r.t $\boldsymbol{\theta}$. By Lemma B.1 we can write this as Tikhonov regularized problem:

$$\inf_{\boldsymbol{\theta}\geq\mathbf{0}} \quad \frac{1}{2}\sum_{m=1}^{M} \theta_m^{-1} \|\boldsymbol{w}_m\|_2^2 + C\sum_{i=1}^{n} l\big(\langle \boldsymbol{w}, \phi(x_i)\rangle + b,\ y_i\big) + \mu \sum_{m=1}^{M} \theta_m^{p/(2-p)} \ .$$

for a suitable chosen constant $\mu > 0$. Let us ignore the positivity constraint $\boldsymbol{\theta} \geq \mathbf{0}$ for the moment; the above objective is differentiable in $\boldsymbol{\theta}$ for $\boldsymbol{\theta} \neq \mathbf{0}$ so that the optimum is attained for the gradient w.r.t. $\boldsymbol{\theta}$ being zero, i.e.,

$$\forall\, m = 1, \ldots, M: \quad -\frac{1}{2}\theta_m^{-2} \|\boldsymbol{w}_m\|^2 + \mu\, p/(2-p)\, \theta_m^{(p/(2-p))-1} = 0 \ .$$

Solving this for $\boldsymbol{\theta}$, we observe that the optimal $\boldsymbol{\theta}$ is indeed positive and we have the proportionality

$$\forall\, m = 1, \ldots, M: \qquad \theta_m \propto \|\boldsymbol{w}_m\|_2^{2-p} \ .$$

Normalizing $\boldsymbol{\theta}$ to fulfill the constraint $\sum_m \theta_m^{p/(2-p)} = 1$ (which is possible because $\boldsymbol{w} \neq \mathbf{0}$) yields the first part of the proposition. Note that $\theta_m \neq 0$ is no restriction as $\theta_m = 0$ can only be when $\|\boldsymbol{w}_m\|_2 = \mathbf{0}$ so that the proposition trivially holds in that case.

For the second part, we proceed similarly but use the dual formulation as a starting point: we write

$$\inf_{\boldsymbol{\theta}\geq\mathbf{0}} \quad C\sum_{i=1}^{n} l^*\Big(-\frac{\alpha_i}{C}, y_i\Big) + \frac{1}{2}\boldsymbol{\alpha}^\top \sum_{m=1}^{M} \theta_m K_m \boldsymbol{\alpha} + \mu \sum_{m=1}^{M} \theta_m^{p/(2-p)}$$

so that setting the gradient w.r.t. $\boldsymbol{\theta}$ to zero yields

$$\forall\, m = 1, \ldots, M: \quad -\frac{1}{2}\boldsymbol{\alpha}^\top K_m \boldsymbol{\alpha} + \mu\, p/(2-p)\, \theta_m^{(p/(2-p))-1} = 0\ .$$

Solving this for $\boldsymbol{\theta}$ results in the proportionality

$$\forall\, m = 1, \ldots, M: \quad \theta_m \propto \left(\boldsymbol{\alpha}^\top K_m \boldsymbol{\alpha}\right)^{-\frac{2-p}{2-2p}}\ .$$

Again, we observe that the so-obtained $\boldsymbol{\theta}$ is nonnegative and normalizing $\boldsymbol{\theta}$ to fulfill the constraint $\sum_m \theta_m^{p/(2-p)} = 1$ (which is possible because $\exists m: K_m \succ 0$) yields the second part of the proposition. ■

We now have all ingredients to formulate a simple macro-wrapper algorithm for ℓ_p-norm MKL training:

Algorithm 3.1 (Analytical Wrapper). Simple ℓ_p-norm MKL training algorithm. SVM computations and analytical updates of $\boldsymbol{\theta}$ are alternated.

```
1: input: p ∈ [1,∞]\{2}
2: For all m initialize θ_m := (1/M)^{(2−p)/p}
3: while optimality conditions are not satisfied do
4:     Compute α := arg(SVM(K_θ))
5:     if p ∈ [1,2[
6:         For all m compute ‖w_m‖ according to Eq. (II.5)
7:     end if
8:     Update θ according to Prop. 3.1
9: end while
10: output: α and θ as sparse vectors
```

The above algorithm starts with initializing $\boldsymbol{\theta}$ uniformly (Line 2) and then alternates between training an SVM in the dual using the actual kernel mixture $K_{\boldsymbol{\theta}}$ (Line 4) and updating $\boldsymbol{\theta}$ (Line 8). Thereby, if $p \in [1, 2[$, the $\boldsymbol{\theta}$-update is performed in the primal using $\boldsymbol{w}$ computed from $\boldsymbol{\alpha}$ by (II.5) (Line 6) and, if $p \in]2, \infty]$, the update is performed in the dual. The algorithm can be stopped when, for example, the duality gap or the change in objective function within subsequent iterations is less than a pre-specified threshold.

Beside its simplicity, an advantage of the above algorithm is its modular form which allows to use existing (efficient) SVM solvers in the $\boldsymbol{\alpha}$-step.

3.2 Large-Scale Algorithm

The above wrapper algorithm computes a full-blown SVM in each iteration. This can be disadvantageous because it is likely that much computational time is spent on suboptimal mixtures. Certainly, suboptimal $\boldsymbol{\alpha}$-solutions would already suffice to improve far-from-optimal $\boldsymbol{\theta}$ in the $\boldsymbol{\theta}$-step.

This is becoming a very pressing problem especially in large-scale machine learning applications.[1] As a remedy, we propose the following algorithm for large-scale MKL optimization:

Algorithm 3.2 (Analytical Chunking). ℓ_p-Norm MKL chunking-based training algorithm via analytical update. The variables $\boldsymbol{\theta}$ and (signed) $\boldsymbol{\alpha}$ are optimized interleavingly. The algorithm is stated here for the hinge loss.

1: **input:** $p \in [1,\infty]\backslash\{2\}$, $Q \in \mathbb{N}$, $\epsilon > 0$
2: **initialize:** $\forall\, i,m:\ g_{m,i} = \hat{g}_i = \alpha_i = 0$; $L = S = -\infty$; $\theta_m := (1/M)^{(2-p)/p}$
3: **iterate**
4: Select l variables $\alpha_{i_1},\dots,\alpha_{i_l}$ based on the gradient $\hat{\mathbf{g}}$ of SVM
5: Store $\boldsymbol{\alpha}^{\text{OLD}} = \boldsymbol{\alpha}$ and then compute $\boldsymbol{\alpha} := \arg(\text{SVM}(K_{\boldsymbol{\theta}}))$ w.r.t. the selected variables
6: Update gradient $\forall\, i,m:\ g_{m,i} := g_{m,i} + \sum_{q=1}^{Q}(\alpha_{i_q} - \alpha_{i_q}^{\text{OLD}})k_m(x_{i_q}, x_i)$
7: Compute the quadratic terms $\forall m:\ S_m := \frac{1}{2}\sum_i g_{m,i}\alpha_i$, $\|\boldsymbol{w}_m\|_2^2 := 2\theta_m^2 S_m$
8: $L_{\text{OLD}} = L$, $L = \sum_i y_i\alpha_i$, $S_{\text{OLD}} = S$, $S = \sum_m \theta_m S_m$
9: **if** $|1 - (L-S)/(L_{\text{OLD}} - S_{\text{OLD}})| \geq \varepsilon$
10: Update $\boldsymbol{\theta}$ according to Prop. 3.1
11: **if** $p \in [1,2]$
12: For all m compute $\|\boldsymbol{w}_m\|$ according to Eq. (II.5)
13: **end if**
14: **else**
15: **break**
16: **end if**
17: $\hat{g}_i = \sum_m \theta_m g_{m,i}$ for all $i = 1,\dots,n$
18: **output:** $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ as sparse vectors

The above algorithm starts with initializing $\boldsymbol{\theta}$ uniformly (Line 2) and then alternates between (incomplete) $\boldsymbol{\alpha}$- and $\boldsymbol{\theta}$-steps. Thereby, the $\boldsymbol{\alpha}$-step is given by standard chunking-based SVM computations and carried out by the SVM$^{\text{light}}$ module in SHOGUN (Lines 4-6). SVM-objective values are computed in Lines 7-8. Finally, the analytical $\boldsymbol{\theta}$-step is carried out in Lines 10-13. The algorithm terminates (Line 15) if the maximal KKT violation (c.f. Joachims, 1999) falls below a predetermined precision ε and if it holds $|1 - \frac{\omega}{\omega_{\text{OLD}}}| < \varepsilon_{\text{MKL}}$ for the normalized maximal constraint violation in the MKL-step, where ω denotes the MKL objective function value.

The main idea of the above algorithm is to perform each $\boldsymbol{\alpha}$-step solely with respect to a small number Q of active variables. Thereby, Q is chosen as described in Joachims (1999); in our computational experiments (see Section 3.4), we observed $Q = 40$ as a typical value.

[1] We refer to machine learning applications as being *large-scale* when the data cannot be stored in memory or the computation reaches a maintainable limit. Note that in the case of MKL this can be due to both a large sample size or a high number of kernels.

3.2.1 Convergence Proof

In this section, we prove the convergence of Algorithm 3.1. To this aim, we invoke a result from Bertsekas (1999) on the convergence of the block coordinate descent method:

Proposition 3.2 (Bertsekas, 1999, p. 268–269)**.** *Let $\mathcal{S} = \bigotimes_{m=1}^{M} \mathcal{S}_m$ be the Cartesian product of closed convex sets $\mathcal{S}_m \subset \mathbb{R}^{d_m}$, be $f : \mathcal{S} \to \mathbb{R}$ a continuously differentiable function. Define the block coordinate descent method recursively by letting $\boldsymbol{s}^0 \in \mathcal{S}$ be any feasible point, and be*

$$\boldsymbol{s}_m^{k+1} = \operatorname{argmin}_{\boldsymbol{\xi} \in \boldsymbol{s}_m} f\left(\boldsymbol{s}_1^{k+1}, \cdots, \boldsymbol{s}_{m-1}^{k+1}, \boldsymbol{\xi}, \boldsymbol{s}_{m+1}^{k}, \cdots, \boldsymbol{s}_M^{k}\right), \quad \forall m = 1, \ldots, M. \tag{3.1}$$

Suppose that for each m and $\boldsymbol{s} \in \mathcal{S}$, the minimum

$$\min_{\boldsymbol{\xi} \in \mathcal{S}_m} f\left(\boldsymbol{s}_1, \cdots, \boldsymbol{s}_{m-1}, \boldsymbol{\xi}, \boldsymbol{s}_{m+1}, \cdots, \boldsymbol{s}_M\right) \tag{3.2}$$

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

The next proposition establishes convergence of the proposed ℓ_p-norm MKL training algorithm 3.1.

Theorem 3.3. *Let l be the hinge loss and be $p \in]1, \infty] \backslash \{2\}$. Let the kernel matrices $K_1, \ldots, K_M$ be strictly positive-definite. Then every limit point of Algorithm 3.1 is a globally optimal point of Problem II.4.*

Proof Note that Algorithm 3.1 can be interpreted a block coordinate descent algorithm, so we write Problem II.4 as a minimization problem

$$\text{if } p \in [1, 2[: \qquad \inf_{(\boldsymbol{w}, b), \boldsymbol{\theta}} \quad \frac{1}{2} \sum_{m=1}^{M} \theta_m^{-1} \left\|\boldsymbol{w}_m\right\|_2^2 + C \sum_{i=1}^{n} l\big(\langle \boldsymbol{w}, \phi(x_i) \rangle + b,\ y_i\big) \tag{3.3}$$

$$\text{if } p \in]2, \infty]: \qquad \inf_{\boldsymbol{\alpha}, \boldsymbol{\theta}} \quad C \sum_{i=1}^{n} l^*\Big(-\frac{\alpha_i}{C}, y_i\Big) + \frac{1}{2} \boldsymbol{\alpha}^\top \sum_{m=1}^{M} \theta_m K_m \boldsymbol{\alpha}\ . \tag{3.4}$$

subject to $\sum_m \theta_m^{p/(2-p)} \leq 1$ and $\boldsymbol{\theta} \geq \mathbf{0}$. In the following, we differentiate the two cases $p \in [1, 2[$ and $p \in]2, \infty]$.

1. The case $p \in [1, 2[$. In the first case, we have to transform (3.3) into a form such that the requirements for the application of Prop. 3.2 are fulfilled. We start by expanding the hinge loss $l(t) = \max(0, 1 - t)$ in (3.3) into

$$\begin{aligned} \min_{\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\theta}} \quad & C \sum_{i=1}^{n} \xi_i + \frac{1}{2} \sum_{m=1}^{M} \theta_m^{-1} \left\|\boldsymbol{w}_m\right\|_2^2, \\ \text{s.t.} \quad & \forall i: \ \sum_{m=1}^{M} \langle \boldsymbol{w}_m, \phi_m(x_i) \rangle_{\mathcal{H}_m} + b \geq 1 - \xi_i; \quad \boldsymbol{\xi} \geq 0; \quad \left\|\boldsymbol{\theta}\right\|_{p/(2-p)} \leq 1; \quad \boldsymbol{\theta} \geq \mathbf{0}, \end{aligned}$$

thereby extending the second block of variables $(\boldsymbol{w}, b)$ to $(\boldsymbol{w}, b, \boldsymbol{\xi})$. Moreover, we note that after an application of the representer theorem[2] (Kimeldorf and Wahba, 1971) we may, without loss of generality, assume $\mathcal{H}_m = \mathbb{R}^n$.

In the problem's current form, the possibility of $\theta_m = 0$ while $\|\boldsymbol{w}_m\| \neq 0$ renders the objective function nondifferentiable. This hinders the application of Prop. 3.2. However, such a pair $(\boldsymbol{w}_m, \theta_m)$ would yield an infinite objective and thus cannot be optimal. Moreover, even $\|\boldsymbol{w}_m\|$=0 would, for $p > 1$, contradict the KKT condition (II.6) because the kernels are assumed to be strictly positive-definite; in any case it thus holds $\theta_m \neq 0$ (given $p > 1$). Therefore we can substitute the constraint $\boldsymbol{\theta} \geq \mathbf{0}$ by $\boldsymbol{\theta} > \mathbf{0}$ for all m without changing the optimal solution. In order to maintain the closeness of the feasible set we subsequently apply a bijective coordinate transformation $\log : \mathbb{R}_+^M \to \mathbb{R}^M$ so that $\theta_m^{\text{NEW}} := \log(\theta_m^{\text{OLD}})$, resulting in the following equivalent problem,

$$\begin{aligned} \inf_{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\theta}} \quad & C\sum_{i=1}^{n}\xi_i + \frac{1}{2}\sum_{m=1}^{M}\exp(-\theta_m)\|\boldsymbol{w}_m\|_2^2, \\ \text{s.t.} \quad & \forall i: \ \sum_{m=1}^{M}\langle \boldsymbol{w}_m, \phi_m(x_i)\rangle + b \geq 1 - \xi_i; \quad \boldsymbol{\xi} \geq 0; \quad \|\exp(\boldsymbol{\theta})\|_{p/(2-p)} \leq 1, \end{aligned} \tag{3.5}$$

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

Applying the block coordinate descent method of Eq. (3.1) to the base problem (3.3) and to the reparameterized problem (3.5) yields the same sequence of solutions $\{(\boldsymbol{w}, b, \boldsymbol{\theta})^k\}_{k\in\mathbb{N}_0}$. The reparameterized problem now allows to apply Prop. 3.2 for the two blocks of coordinates $\boldsymbol{\theta} \in \mathcal{S}_1 := \mathbb{R}$ and $(\boldsymbol{w}, b, \boldsymbol{\xi}) \in \mathcal{S}_2 := \mathbb{R}^{n(M+1)+1}$: the objective is continuously differentiable and the sets $\mathcal{S}_1$ and $\mathcal{S}_2$ are closed and convex. To see the latter, note that $\|\cdot\|_{p/(2-p)} \circ \exp$ is a convex function, since $\|\cdot\|_{p/(2-p)}$ is convex and non-increasing in each argument (cf., e.g., Section 3.2.4 in Boyd and Vandenberghe, 2004). Moreover, the minima in Eq. (3.1) 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 (3.3). For a convex problem, this is equivalent to such a limit point being globally optimal.

2. The case $p \in]2, \infty]$. In the second case there is nothing to do since for hinge loss (3.4) trivially fulfills the conditions for the application of Proposition 3.2. ■

3.2.2 Practical Considerations

In practice, we are facing two problems. First, the standard Hilbert space setup necessarily implies that $\|\boldsymbol{w}_m\| \geq 0$ for all m. In practice, this assumption may often be

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

violated, either due to numerical imprecision or because of using an indefinite "kernel" function. However, for any $\|\boldsymbol{w}_m\| \leq 0$ it 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 θ_m are then computed according to Prop. 3.1 but with the normalization being adjusted accordingly.

Second, in practice, the SVM problem will only be solved up to finite precision, which may lead to convergence problems. Moreover, we want to improve $\boldsymbol{\alpha}$ only a little bit before recomputing $\boldsymbol{\theta}$ since computing a high precision solution can be wasteful (cf. Alg. 3.2). We can obtain a heuristic to overcome the potential convergence problem by computing the SVM by a higher precision if needed. The main idea is that this way it is likely that the primal objective decreases within each $\boldsymbol{\alpha}$-step. 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 in our computational experiments (shown later in this thesis).

Finally, we note that, of course, for linear kernels the SVM-step of the proposed block coordinate descent algorithm could also be carried out in the primal. In the view of efficient primal SVM optimizers such as LibLinear (Fan et al., 2008) or Ocas (Franc and Sonnenburg, 2008) this is extremely appealing.

3.3 Implementation

Both of the algorithms described in the previous section were implemented in C++ for regression, one-class classification, and two-class classification tasks. The implementation has been made available as a part of the open source machine learning toolbox SHOGUN (Sonnenburg et al., 2010). SHOGUN, which contains interfaces to MATLAB, Octave, Phyton, and R, can be freely obtained from `http://www.shogun-toolbox.org`.

How to use?

The user can choose from a variety of options. First, in contrast to prevalent MKL implementations, not only precomputed kernels can be used as an input to our software but kernels can also be computed on-the-fly during operation (in that case kernel caching is employed); on-the-fly kernel computation can be very appealing if the kernel matrix is too large to be held in memory, as commonly encountered in large-scale machine learning applications. Second, one can choose the MKL optimization algorithm to be used—valid options are any of the proposed analytical optimization schemes and the cutting plane strategy described in the appendix. In case Algorithm 3.1 is used, the $\boldsymbol{\alpha}$-step can be carried out by any of the SVM implementations contained in SHOGUN (this includes, among many others, LIBSVM, LIBLINEAR, and SVM$^{\text{LIGHT}}$). The cutting plane optimizer requires IBM ILOG CPLEX[3]; alternatively, but only for $p = 1$,

[3] `http://www.ibm.com/software/integration/optimization/cplex/`.

the free LP solver GLPK[4] can be chosen.

Implementation Details

Both, the cutting plane strategy and the analytical algorithm (Algorithm 3.2) perform interleaved optimization and thus require modification of the core SVM optimization algorithm. This is currently integrated into the chunking-based SVRlight and SVMlight. To reduce the implementation effort, we implement a single function `perform_mkl_step(lin, quad)`, that has two arguments: first, the linear term $\texttt{lin} := \sum_{i=1}^{n} \alpha_i$ and second, an array $\texttt{quad} = \left(\texttt{quad}_m\right)_{m=1}^{M}$ containing the quadratic terms of the respective kernels $\texttt{quad}_m := \frac{1}{2}\boldsymbol{\alpha}^T K_m \boldsymbol{\alpha}$. This function is either, in the interleaved optimization case, called as a callback function (after each chunking step), or it is called by the wrapper algorithm (after each SVM optimization to full precision).

One-class classification is trivially implemented using $\texttt{lin} = 0$ and 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 $\texttt{lin} = -\sum_{i=1}^{n}(\alpha_i + \alpha_i^*)\varepsilon - \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)y_i$. Since this requires modification of the core SVM solver, we implement SVR only for interleaved optimization and SVMlight.

Note that the choice of the size of the kernel cache becomes crucial when applying MKL to large-scale learning applications. While conventional wrapper algorithms only solve a *single* kernel SVM in each iteration and thus use a single large kernel cache, interleaved optimization methods must keep track of several partial MKL objectives obj_m and thus require access to individual kernel rows. Therefore, the same cache size should be used for all sub-kernels.

3.3.1 Kernel Normalization

In practice, 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 so that they penalize those features less (more). We here use isotropic regularizers, which penalize all dimensions uniformly. This implies that the kernels have to be normalized in a sensible way in order to represent an "uninformative prior" as to which kernels are useful.

There exist several approaches to kernel normalization, of which we use two in the computational experiments below. They are fundamentally different. The first one generalizes the common practice of standardizing features to entire kernels, thereby directly

[4] `http://www.gnu.org/software/glpk/`.

implementing the spirit of the discussion above. In contrast, the second normalization approach rescales the data points to unit norm in feature space. Nevertheless, it can have a beneficial effect on the scaling of kernels, as we argue below.

Both normalization approaches, as well as a number of additional ones, were implemented and made available as a part of our SHOGUN implementation (see SHOGUN documentation for a complete list of normalization methods).

Method 1: Multiplicative Normalization

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

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

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

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

so that the final normalization rule is

$$k(x,\bar{x}) \longmapsto \frac{k(x,\bar{x})}{\frac{1}{n}\sum_{i=1}^{n}k(x_i,x_i) - \frac{1}{n^2}\sum_{i,j=1}^{n},k(x_i,x_j)}. \tag{3.6}$$

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

Method 2: Spherical Normalization

Another approach, which is frequently applied in the MKL literature, is to normalize

$$k(x,\bar{x}) \longmapsto \frac{k(x,\bar{x})}{\sqrt{k(x,x)k(\bar{x},\bar{x})}}. \tag{3.7}$$

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

Thus, spherical normalization may be seen as an approximation of 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 (for example, in text classification, the category of a text rather depends on its word frequency distribution than on its length so that spherical normalization is desired).

3.4 Runtime Experiments

In this section, we analyze the efficiency of our implementation of ℓ_p-norm MKL. We experiment on the MNIST data set, where the task is to separate odd and even digits. The digits in this $n = 60,000$-elemental data set, which we downloaded from `http://yann.lecun.com/exdb/mnist/`, are of size 28x28, leading to $d = 784$ dimensional examples. We compare the proposed optimization approaches for ℓ_p-norm MKL with the state-of-the-art for ℓ_1-norm MKL, namely, SimpleMKL (Rakotomamonjy et al., 2008), HessianMKL (Chapelle and Rakotomamonjy, 2008), and the cutting plane method (CPM) proposed in Sonnenburg et al. (2006a).[5] We also experiment with the analytical method for $p = 1$, although convergence is only guaranteed by Theorem 3.3 for $p > 1$. In addition, we compare all approaches to a standard single-kernel SVM, namely, SVM$^{\texttt{light}}$ using a uniform kernel mixture (ℓ_2-norm MKL).

We experiment with MKL using precomputed kernels (excluding the kernel computation time from the timings) and MKL based on on-the-fly computed kernel matrices measuring training time including kernel computations. Naturally, runtimes of on-the-fly methods should be expected to be higher than the ones of the precomputed counterparts; on the other hand, on-the-fly methods are not subject to memory constraints while precomputation-based methods eventually run out of memory with increasing n.

We optimize all of our methods up to a precision of $\varepsilon_{\texttt{svm}} = 10^{-3}$ in the $\boldsymbol{\alpha}$-step and $\varepsilon_{\texttt{mkl}} = 10^{-5}$ in the $\boldsymbol{\theta}$-step (those values have proven to be adequate in practice). Subsequently, we compute the relative duality gaps of the ℓ_1-cutting-plane method for each run and input them as stopping criteria of SimpleMKL and HessianMKL. This is to ensure that the comparison is fair (i.e., all methods are optimized to the same precision). The SVM trade-off parameters are set to $C = 1$ for all methods (which is a natural choice; but note that even a different value of C should affect all methods equally as it only affects the $\boldsymbol{\alpha}$-step of the MKL optimizers). The runtime differences between the p-norms were marginal for $p \in]1, \infty[$ so that for non-sparse $\ell_{p>1}$-norm MKL we only plot the results for $p = \frac{4}{3}$ (which is a natural value as it lies right in the middle between the extreme cases ℓ_1-norm MKL and uniform-sum-kernel SVM).

Experiment 1: Runtime in the Number of Training Examples

Figure 3.1 (top) displays the results for varying sample sizes but a fixed number of 50 precomputed or on-the-fly computed Gaussian kernels with bandwidths $2\sigma^2 \in 1.2^{0,\ldots,49}$.

[5] The implementations were obtained from `http://asi.insa-rouen.fr/enseignants/~arakotom/code/` (SimpleMKL) and `http://olivier.chapelle.cc/ams/hessmkl.tgz` (HessianMKL).

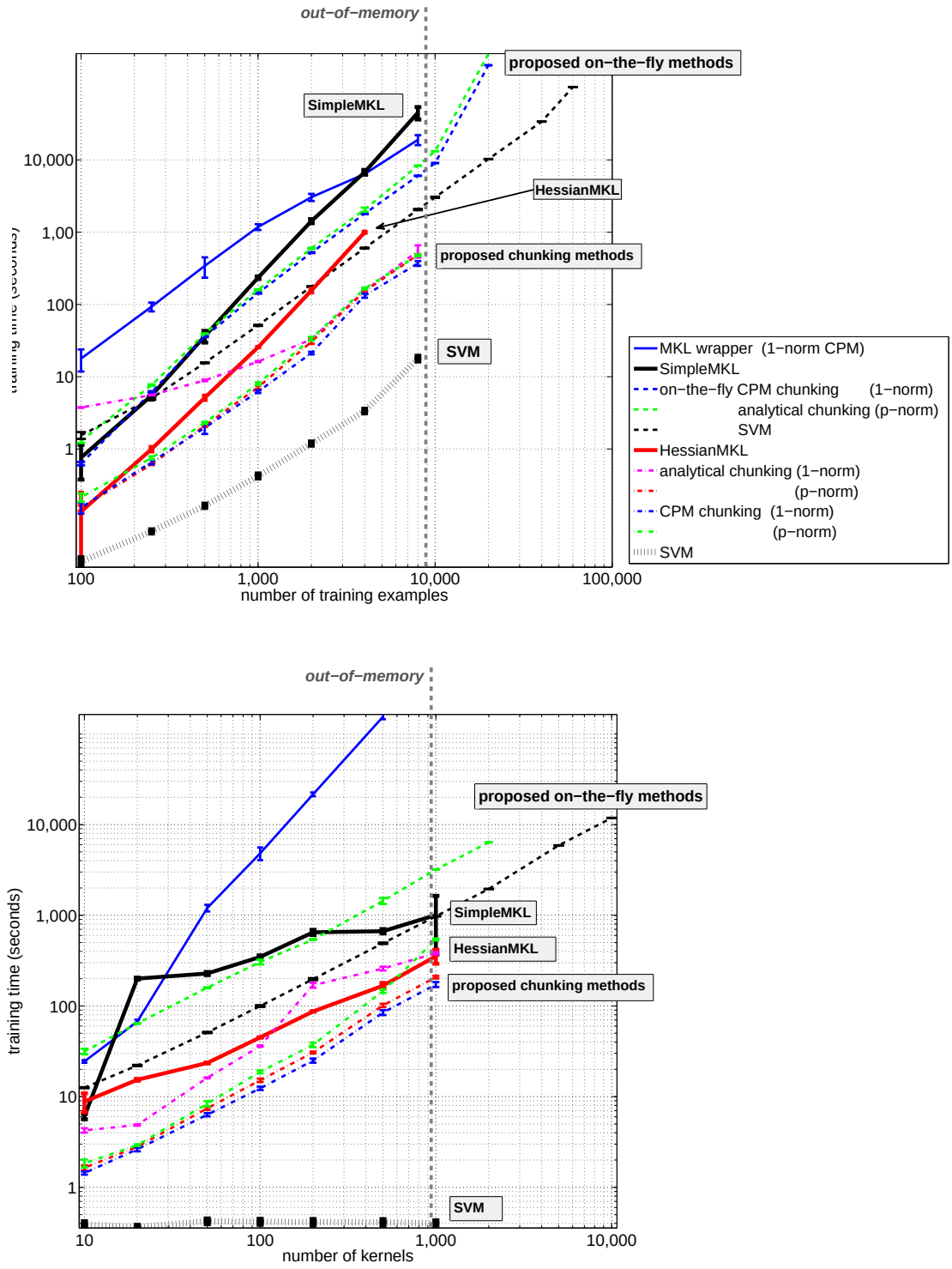


Figure 3.1: Runtime of ℓ_p-norm MKL in the number of training examples (Top) and the number of kernels (Bottom), respectively, and comparison to MKL baselines.

Error bars indicate standard errors over 5 repetitions. As expected, the SVM using a precomputed uniform kernel mixture is the fastest method. The classical MKL-wrapper-based methods, SimpleMKL and the CPM wrapper, are the slowest; they are even slower than methods that compute kernels on-the-fly. Note that the on-the-fly methods naturally have higher runtimes because they do not profit from precomputed kernel matrices.

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

Non-sparse $\ell_{p>1}$-norm MKL scales similarly as the ℓ_1-norm CPM for all chunking-based optimization strategies: analytical and CPM chunking. Naturally, the generalized $\ell_{p>1}$-norm CPM is slightly slower than its ℓ_1-norm counterpart as an additional series of Taylor expansions has to be computed within each $\boldsymbol{\theta}$-step. HessianMKL ranks in between on-the-fly and ℓ_p-norm chunking methods.

Experiment 2: Runtime in the Number of Kernels

Figure 3.1 (bottom) shows the execution times as a function of the number of RBF kernels M. Thereby, the sample size is fixed to $n = 1000$. The bandwidths of the RBF kernels are taken as $2\sigma^2 \in 2^{0,\dots,M-1}$. As expected, the SVM with the unweighted-sum kernel is hardly affected by this setup and essentially has a constant training time. The ℓ_1-norm CPM handles the increasing number of kernels best and is the fastest MKL method. Non-sparse approaches to MKL show reasonable run-times, being just slightly slower. Thereby, the analytical methods are somewhat faster than the CPM approaches. The sparse analytical method performs substantially worse than its non-sparse counterpart; this might be related to the fact that convergence of the analytical method is only guaranteed for $p > 1$. The wrapper methods (CPM wrapper and SimpleMKL) again perform worst.

However, in contrast to the previous experiment, SimpleMKL becomes more efficient with an increasing number of kernels. We conjecture that this is in part owing to the fact that the line search (which in the case of SimpleMKL is a full-blown SVM computation) is relatively fast for moderate sample sizes. Nevertheless, the capacity of SimpleMKL remains limited due to memory restrictions of the hardware. For example, for storing 1,000 kernel matrices for 1,000 data points, about 7.4GB of memory are required. On the other hand, our interleaved optimizers which allow for effective caching can easily cope with 10,000 kernels of the same size (74GB). HessianMKL is considerably faster than SimpleMKL but slower than the $\ell_{p>1}$-norm chunking-based methods and the ℓ_1-norm CPM; similar to SimpleMKL, it becomes more efficient with an increasing number of kernels but eventually runs out of memory.

Overall, the proposed chunking-based optimization strategies (analytic and CPM) achieve a speedup of up to one and two orders of magnitude over HessianMKL and SimpleMKL, respectively. Using efficient kernel caching, they allow for truly large-scale multiple kernel learning—well beyond the limits imposed by having to precompute and store the complete kernel matrices.

3.5 Summary and Discussion

In this chapter, we derived two efficient algorithms for solving the ℓ_p-norm MKL optimization problem. Both algorithms were, for the sake of performance, implemented in C++ and made available as a part of the open source machine learning toolbox SHOGUN (Sonnenburg et al., 2010), which also contains interfaces to MATLAB, Octave, Phyton, and R. In our computational experiments, we found these large-scale optimization algorithms allowing us to deal with ten thousands of data points and thousands of kernels at the same time, as demonstrated on the MNIST data set. We compared our algorithms to the state-of-the-art in MKL research, namely HessianMKL (Chapelle and Rakotomamonjy, 2008) and SimpleMKL (Rakotomamonjy et al., 2008), and found ours to be up to two magnitudes faster. Both algorithms are based on the alternative, wrapper-based ℓ_p-norm MKL formulation (Problem II.4) in contrast to the original problem (Problem II.3).

The first algorithm, the analytical wrapper, consists of an iterate 2-step procedure, alternatingly performing the analytical MKL step and calling an SVM solver (to this end, both, SVMlight and LIBSVM, are implemented in SHOGUN). Clearly, AnalyticalMKL is even simpler than SimpleMKL since the latter requires a heuristic line search in the MKL step. We showed AnalyticalMKL being provably convergent, using the usual regularity assumptions. The second algorithm, the analytical chunking, is directly integrated into the SVM light code (i.e., the MKL step is called after each chunking iteration), allowing for truly large-scale MKL.

We remark that the proposed algorithms can, in particular, be used to optimize the classical $\ell_{p=1}$-norm MKL formulation. However, we found the algorithms in some cases to converge considerably faster in the non-sparse case $p > 1$, where the smoothness of the objective can be exploited. Considering the gain in prediction accuracy that is also achieved by $\ell_{p>1}$-norm MKL, established later in this dissertation, this might be remarkable. Note that we also wrote scripts completely automating the whole process from training over model selection to evaluation. Currently, MKL can be trained and validated by a single line of MATLAB code including random subsampling, model search for the optimal parameters C and p, and collection of results. Our software is freely available under the GPL license. We also remark that another fast algorithm based on cutting planes is described in Appendix D. We found this algorithm to be even faster in some cases but it requires a commercial QCQP solver so that it cannot be released under the GPL license.

Finally, we note that performing MKL with 1,000 precomputed kernel matrices of

size 1,000 × 1,000 requires less than 3 minutes for the chunking methods. This suggests to focus future research efforts rather on improving the *accuracy* of MKL models rather than accelerating the optimization algorithms.

4 Theoretical Analysis

In this chapter, we derive upper bounds on the global and local Rademacher complexities of ℓ_p-norm MKL, from which we deduce excess risk bounds that achieve fast convergence rates of the order $O(n^{-\frac{\alpha}{1+\alpha}})$, where α is the minimum eigenvalue decay rate of the individual kernels (previous bounds for ℓ_p-norm MKL only achieved $O(n^{-\frac{1}{2}})$). We also give a lower bound that, up to constants, matches the upper bounds, showing that our results are tight. Finally, the generalization performance of ℓ_p-norm MKL, as guaranteed by the excess risk bound, is studied for varying values of p, shedding light on why learning kernels can help performance. This chapter is based on mathematical preliminaries introduced in Appendix A.

The main **contributions** in this chapter are the following:

- We derive an upper bound on the local Rademacher complexity of ℓ_p-norm multiple kernel learning that yields an excess risk bound achieving fast convergence rates, while previous approaches only yielded slow rates.
- We also prove a lower bound that matches the upper one, showing that our result is tight.
- As a byproduct, we extend a previous upper bound of Cortes et al. (2010a) on the *global* Rademacher complexity to the full range of $p \in [1, \infty]$, yet presenting a substantially easier proof and improved constants.
- We exemplarily evaluate the bound for several scenarios that differ in the soft sparsity of the underlying Bayes hypothesis, shedding light on why learning kernels can help performance.

Parts of this chapter are based on:

M. Kloft and G. Blanchard. The local Rademacher Complexity of Multiple Kernel Learning. *ArXiv preprint 1103.0790v1.* Submitted to *Journal of Machine Learning Research (JMLR)* , Mar 2011.

A short version has been submitted to *Twenty-Fifth Annual Conference on Neural Information Processing Systems (NIPS 2011)*, Jun 2011.

Notation

For the upcoming analysis it is convenient to view MKL as an empirical minimization algorithm acting on the hypothesis class

$$H_{p,D,M} = \left\{ f_{\boldsymbol{w}} : x \mapsto \langle \boldsymbol{w}, \phi(x) \rangle \;\middle|\; \boldsymbol{w} = (\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(M)}), \|\boldsymbol{w}\|_{2,p} \leq D \right\} \tag{4.1}$$

(if D or M are clear from the context, we sometimes denote $H_p = H_{p,D} = H_{p,D,M}$). This is equivalent to the original formulation in Problem 2.1, where we viewed MKL as regularized risk minimization problem. To see the equivalence between the regularization and hypothesis-class views, note that we can equivalently translate the Tikhonov-regularized MKL problem in an Ivanov-regularized problem by applying Lemma B.1 (presented in Appendix B).

Furthermore, let P be a probability measure on $\mathcal{X}$ independently generating the data $x_1, \ldots, x_n$ and denote by $\mathbb{E}$ the corresponding expectation operator. Note that for $x \sim P$ we can view $\phi(x)$ and $\phi_m(x)$ as random variables taking values in the Hilbert spaces $\mathcal{H}$ and $\mathcal{H}_m$, respectively. Correspondingly, we will work with covariance operators in Hilbert spaces. In a finite dimensional vector space, the (uncentered) covariance operator can be defined in the usual vector/matrix notation as $\mathbb{E}\phi(x)\phi(x)^\top$. Since we are working with potentially infinite-dimensional vector spaces, instead of $\phi(x)\phi(x)^\top$ we will use the tensor notation $\phi(x) \otimes \phi(x) \in \mathrm{HS}(\mathcal{H})$, which is a Hilbert-Schmidt operator $\mathcal{H} \mapsto \mathcal{H}$ defined as $(\phi(x) \otimes \phi(x))\boldsymbol{u} = \langle \phi(x), \boldsymbol{u} \rangle \phi(x)$. The space $\mathrm{HS}(\mathcal{H})$ of Hilbert-Schmidt operators on $\mathcal{H}$ is itself a Hilbert space, and the expectation $\mathbb{E}\phi(x) \otimes \phi(x)$ is well-defined and belongs to $\mathrm{HS}(\mathcal{H})$ as soon as $\mathbb{E}\,\|\phi(x)\|^2$ is finite, which will always be assumed as such (as a matter of fact, we will often assume that $\|\phi(x)\|$ is bounded a.s.). We denote by $J = \mathbb{E}\phi(x) \otimes \phi(x)$ and $J_m = \mathbb{E}\phi_m(x) \otimes \phi_m(x)$ the uncentered covariance operators corresponding to variables $\phi(x)$ and $\phi_m(x)$, respectively; it holds that $\mathrm{tr}(J) = \mathbb{E}\,\|\phi(x)\|_2^2$ and $\mathrm{tr}(J_m) = \mathbb{E}\,\|\phi_m(x)\|_2^2$.

4.1 Global Rademacher Complexity

We first review global Rademacher complexities (GRC) in multiple kernel learning. Let $x_1, \ldots, x_n$ be an i.i.d. sample drawn from P. The global Rademacher complexity is defined as $R(H_p) = \mathbb{E} \sup_{f_{\boldsymbol{w}} \in H_p} \langle \boldsymbol{w}, \frac{1}{n} \sum_{i=1}^n \sigma_i \phi(x_i) \rangle$, where $(\sigma_i)_{1 \le i \le n}$ is an i.i.d. family (independent of $\phi(x_i)$) of Rademacher variables (random signs). Its empirical counterpart is denoted by $\widehat{R}(H_p) = \mathbb{E}\big[R(H_p)\big|x_1, \ldots, x_n\big] = \mathbb{E}_{\boldsymbol{\sigma}} \sup_{f_{\boldsymbol{w}} \in H_p} \langle \boldsymbol{w}, \frac{1}{n} \sum_{i=1}^n \sigma_i \phi(x_i) \rangle$. The interest in the global Rademacher complexity comes from the fact that, if known, it can be used to bound the generalization error (Koltchinskii, 2001; Bartlett and Mendelson, 2002); see Appendix A.

In the recent paper of Cortes et al. (2010a) it was shown, using a combinatorial argument, that the empirical version of the global Rademacher complexity can be bounded as

$$\widehat{R}(H_p) \le \frac{D}{n} \sqrt{cp^* \Big\| \big(\mathrm{tr}(K_m)\big)_{m=1}^M \Big\|_{\frac{p^*}{2}}} \qquad \text{(for } p \in [1,2] \text{ and } p^* \text{ being an integer)}$$

where $c = \frac{23}{22}$. We will now show a quite short proof of this result (extending it to the whole range $p \in [1, \infty]$) and then present a novel bound on the population version of the GRC. The proof presented here is based on the Khintchine-Kahane inequality (Kahane, 1985) using the constants taken from Lemma 3.3.1 and Proposition 3.4.1 in Kwapién and Woyczyński (1992).

Lemma 4.1 (Khintchine-Kahane inequality)**.** *Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_M \in \mathcal{H}$. Then, for any $q \ge 1$, it holds*

$$\mathbb{E}_{\boldsymbol{\sigma}} \Big\| \sum_{i=1}^n \sigma_i \boldsymbol{v}_i \Big\|_2^q \le \Big(c \sum_{i=1}^n \big\| \boldsymbol{v}_i \big\|_2^2 \Big)^{\frac{q}{2}},$$

where $c = \max(1, p^* - 1)$. In particular, the result holds for $c = p^*$.

Proposition 4.2 (Global Rademacher complexity bound, empirical version)**.** *For any $p \geq 1$ the empirical version of global Rademacher complexity of the multi-kernel class H_p can be bounded as*

$$\widehat{R}(H_p) \leq \min_{t \in [p,\infty]} D \sqrt{\frac{t^*}{n} \Big\| \Big(\frac{1}{n} \mathrm{tr}(K_m) \Big)_{m=1}^{M} \Big\|_{\frac{t^*}{2}}}.$$

Proof First note that it suffices to prove the result for $t = p$ as trivially $\|\boldsymbol{w}\|_{2,t} \leq \|\boldsymbol{w}\|_{2,p}$ holds for all $t \geq p$ so that $H_p \subseteq H_t$ and therefore $R(H_p) \leq R(H_t)$. We can use a block-structured version of Hölder's inequality (cf. Lemma C.1) and the Khintchine-Kahane (K.-K.) inequality (cf. Lemma 4.1) to bound the empirical version of the global Rademacher complexity as follows:

$$\begin{aligned}
\widehat{R}(H_p) &\overset{\text{def.}}{=} \mathbb{E}_{\boldsymbol{\sigma}} \sup_{f_{\boldsymbol{w}} \in H_p} \langle \boldsymbol{w}, \frac{1}{n} \sum_{i=1}^{n} \sigma_i \phi(x_i) \rangle \\
&\overset{\text{Hölder}}{\leq} \mathbb{E}_{\boldsymbol{\sigma}} \sup_{f_{\boldsymbol{w}} \in H_p} \|\boldsymbol{w}\|_{2,p} \Big\| \frac{1}{n} \sum_{i=1}^{n} \sigma_i \phi(x_i) \Big\|_{2,p^*} \\
&\overset{(4.1)}{\leq} D \, \mathbb{E}_{\boldsymbol{\sigma}} \Big\| \frac{1}{n} \sum_{i=1}^{n} \sigma_i \phi(x_i) \Big\|_{2,p^*} \\
&\overset{\text{Jensen}}{\leq} D \Big(\mathbb{E}_{\boldsymbol{\sigma}} \sum_{m=1}^{M} \Big\| \frac{1}{n} \sum_{i=1}^{n} \sigma_i \phi_m(x_i) \Big\|_2^{p^*} \Big)^{\frac{1}{p^*}} \\
&\overset{\text{K.-K.}}{\leq} D \sqrt{\frac{p^*}{n}} \Big(\sum_{m=1}^{M} \Big(\underbrace{\frac{1}{n} \sum_{i=1}^{n} \big\| \phi_m(x_i) \big\|_2^2}_{= \frac{1}{n} \mathrm{tr}(K_m)} \Big)^{\frac{p^*}{2}} \Big)^{\frac{1}{p^*}} \\
&= D \sqrt{\frac{p^*}{n} \Big\| \Big(\frac{1}{n} \mathrm{tr}(K_m) \Big)_{m=1}^{M} \Big\|_{\frac{p^*}{2}}},
\end{aligned}$$

which is what was to be shown. ■

Remark. Note that there is a very good reason to state the above bound in terms of $t \geq p$ instead of solely in terms of p: the Rademacher complexity $\widehat{R}(H_p)$ is not monotonic in p and thus it is not always the best choice to take $t := p$ in the above bound. This can be readily seen, for example, for the simple case where all kernels have the same trace—in that case the bound translates into $\widehat{R}(H_p) \leq \frac{D}{n} \sqrt{t^* M^{\frac{2}{t^*}} \mathrm{tr}(K_1)}$. Interestingly, the function $s \mapsto s M^{2/s}$ is not monotone and attains its minimum for $s = 2 \log M$, where log denotes the natural logarithm with respect to the base e. This has interesting consequences: for any $p \leq (2 \log M)^*$ we can take the bound $\widehat{R}(H_p) \leq \frac{D}{n} \sqrt{e \log(M) \mathrm{tr}(K_1)}$, which has only a mild dependancy on the number of

kernels; note that, in particular, we can take this bound for the ℓ_1-norm class $\widehat{R}(H_1)$ for all $M > 1$.

Despite the simplicity of the above proof, the constants are a little better than the ones achieved in Cortes et al. (2010a). However, computing the population version of the global Rademacher complexity of MKL is somewhat more involved and, to the best of our knowledge, has not been addressed yet by the literature. To this end, note that from the previous proof we obtain $R(H_p) = \mathbb{E}\, D\sqrt{p^*/n}\big(\sum_{m=1}^M \big(\frac{1}{n}\sum_{i=1}^n \big\|\phi_m(x_i)\big\|_2^2\big)^{\frac{p^*}{2}}\big)^{\frac{1}{p^*}}$. We can thus use Jensen's inequality to move the expectation operator inside the root,

$$R(H_p) = D\sqrt{p^*/n}\Big(\sum_{m=1}^M \mathbb{E}\big(\frac{1}{n}\sum_{i=1}^n \big\|\phi_m(x_i)\big\|_2^2\big)^{\frac{p^*}{2}}\Big)^{\frac{1}{p^*}}, \tag{4.2}$$

but now need a handle on the $\frac{p^*}{2}$-th moments. To this aim, we use the inequalities of Rosenthal (1970) and Young (e.g., Steele, 2004) to show the following Lemma.

Lemma 4.3 (Rosenthal + Young)**.** *Let $X_1, \ldots, X_n$ be independent nonnegative random variables satisfying $\forall i : X_i \le B < \infty$ almost surely. Then, denoting $c_q = (2qe)^q$, for any $q \ge \frac{1}{2}$ it holds*

$$\mathbb{E}\bigg(\frac{1}{n}\sum_{i=1}^n X_i\bigg)^q \le c_q\bigg(\Big(\frac{B}{n}\Big)^q + \Big(\frac{1}{n}\sum_{i=1}^n \mathbb{E}X_i\Big)^q\bigg).$$

The proof is deferred to Appendix C. It is now easy to show:

Corollary 4.4 (Global Rademacher complexity bound, population version)**.** *Assume the kernels are uniformly bounded, that is, $\|k\|_\infty \le B < \infty$, almost surely. Then for any $p \ge 1$ the population version of the global Rademacher complexity of the multi-kernel class H_p can be bounded as*

$$R(H_{p,D,M}) \le \min_{t\in[p,\infty]} D\, t^* \sqrt{\frac{e}{n}\Big\|\big(\mathrm{tr}(J_m)\big)_{m=1}^M\Big\|_{\frac{t^*}{2}}} + \frac{\sqrt{Be}DM^{\frac{1}{t^*}}t^*}{n}.$$

For $t \ge 2$ the right-hand term can be discarded and the result also holds for unbounded kernels.

Proof As in the previous proof it suffices to prove the result for $t = p$. From (4.2) we conclude by the previous Lemma

$$\begin{aligned} R(H_p) &\le D\sqrt{\frac{p^*}{n}}\Bigg(\sum_{m=1}^M (ep^*)^{\frac{p^*}{2}}\bigg(\Big(\frac{B}{n}\Big)^{\frac{p^*}{2}} + \Big(\underbrace{\mathbb{E}\frac{1}{n}\sum_{i=1}^n \big\|\phi_m(x_i)\big\|_2^2}_{=\mathrm{tr}(J_m)}\Big)^{\frac{p^*}{2}}\bigg)\Bigg)^{\frac{1}{p^*}} \\ &\le D\, p^* \sqrt{\frac{e}{n}\Big\|\big(\mathrm{tr}(J_m)\big)_{m=1}^M\Big\|_{\frac{p^*}{2}}} + \frac{\sqrt{Be}DM^{\frac{1}{p^*}}p^*}{n}, \end{aligned}$$

where for the last inequality we use the subadditivity of the root function. Note that for $p \geq 2$ it is $p^*/2 \leq 1$ and thus it suffices to employ Jensen's inequality instead of the previous lemma so that we get by without the last term on the right-hand side. ■

Interpretation

When, for example, the traces of the kernels are bounded, the above bound is essentially determined by $O\Big(\min_{t \geq p} \frac{t^* M^{\frac{1}{t^*}}}{\sqrt{n}}\Big)$, which indicates a dependancy on the number of kernels that is monotone in the norm-parameter p. In the extreme case $p = 1$, by setting $t = (\log(M))^*$, the bound can be more compactly written $R(H_1) = O\Big(\frac{\log M}{\sqrt{n}}\Big)$.

4.2 Local Rademacher Complexity

Let $x_1, \ldots, x_n$ be an i.i.d. sample drawn from P. We define the *local* Rademacher complexity of H_p as $R_r(H_p) = \mathbb{E} \sup_{f_{\boldsymbol{w}} \in H_p : P f_{\boldsymbol{w}}^2 \leq r} \langle \boldsymbol{w}, \frac{1}{n} \sum_{i=1}^n \sigma_i \phi(x_i) \rangle$, where $P f_{\boldsymbol{w}}^2 := \mathbb{E}(f_{\boldsymbol{w}}(\phi(x)))^2$. Note that it subsumes the global RC as a special case for $r = \infty$. Our interest in the local Rademacher complexity stems from the fact that, if known, it can be used to obtain fast convergence rates of the excess risk (see subsequent section).

We will state our local Rademacher bounds in terms of the spectra of the kernels: this is possible since covariance operators (as self-adjoint, positive Hilbert-Schmidt operators) enjoy discrete eigenvalue-eigenvector decompositions $J = \mathbb{E}\phi(x) \otimes \phi(x) = \sum_{j=1}^\infty \lambda_j \boldsymbol{u}_j \otimes \boldsymbol{u}_j$ and $J_m = \mathbb{E} x^{(m)} \otimes x^{(m)} = \sum_{j=1}^\infty \lambda_j^{(m)} \boldsymbol{u}_j^{(m)} \otimes \boldsymbol{u}_j^{(m)}$, where $(\boldsymbol{u}_j)_{j \geq 1}$ and $(\boldsymbol{u}_j^{(m)})_{j \geq 1}$ form orthonormal bases of $\mathcal{H}$ and $\mathcal{H}_m$, respectively.

We will also use the following assumption in the bounds for the case $p \in [1, 2]$:

Assumption (U) (no-correlation). *Let $x \sim P$. The Hilbert space valued random variables $\phi_1(x), \ldots, \phi_M(x)$ are said to be (pairwise) uncorrelated if for any $m \neq m'$ and $\boldsymbol{w} \in \mathcal{H}_m, \boldsymbol{w}' \in \mathcal{H}_{m'}$, the real variables $\langle \boldsymbol{w}, \phi_m(x) \rangle$ and $\langle \boldsymbol{w}', \phi_{m'}(x) \rangle$ are uncorrelated.*

Since $\mathcal{H}_m, \mathcal{H}_{m'}$ are reproducing kernel Hilbert spaces with kernels $k_m, k_{m'}$, if we go back to the input random variable in the original space $x \in \mathcal{X}$, the above property is equivalent to saying that for any fixed $t, t' \in \mathcal{X}$, the variables $k_m(x, t)$ and $k_{m'}(x, t')$ are uncorrelated. This is the case, for example, if the original input space $\mathcal{X}$ is $\mathbb{R}^M$, the original input variable $x \in \mathcal{X}$ has independent coordinates, and the kernels $k_1, \ldots, k_M$ each act on a different coordinate. Such a setting was considered in particular by Raskutti et al. (2010) in the setting of ℓ_1-penalized MKL. We discuss this assumption in more detail in Section 4.3.2.

We are now equipped to state our main results:

Theorem 4.5 (Local Rademacher complexity bound, $p \in [1, 2]$)**.** *Assume that the kernels are uniformly bounded ($\|k\|_\infty \leq B < \infty$) and that Assumption* (U) *holds.*

The local Rademacher complexity of the multi-kernel class H_p can be bounded for any $p \in [1,2]$ as

$$R_r(H_p) \leq \min_{t\in[p,2]} \sqrt{\frac{16}{n}\left\| \left(\sum_{j=1}^{\infty} \min\left(rM^{1-\frac{2}{t^*}}, ceD^2 t^{*2} \lambda_j^{(m)} \right) \right)_{m=1}^{M} \right\|_{\frac{t^*}{2}}} + \frac{\sqrt{Be} D M^{\frac{1}{t^*}} t^*}{n}.$$

Theorem 4.6 (Local Rademacher complexity bound, $p \in [2,\infty]$)**.** *The local Rademacher complexity of the multi-kernel class H_p can be bounded for any $p \in [2,\infty]$ as*

$$R_r(H_p) \leq \min_{t\in[p,\infty]} \sqrt{\frac{2}{n} \sum_{j=1}^{\infty} \min(r, D^2 M^{\frac{2}{t^*}-1} \lambda_j)}.$$

Remark 1. Note that for the case $p = 1$, by using $t = (\log(M))^*$ in Theorem 4.5, we obtain the bound

$$R_r(H_1) \leq \sqrt{\frac{16}{n}\left\| \left(\sum_{j=1}^{\infty} \min\left(rM, e^3 D^2 (\log M)^2 \lambda_j^{(m)} \right) \right)_{m=1}^{M} \right\|_{\infty}} + \frac{\sqrt{B} e^{\frac{3}{2}} D \log(M)}{n},$$

for all $M \geq e^2$ (see below after the proof of Theorem 4.5 for a detailed justification).

Remark 2. The result of Theorem 4.6 (for $p \in [2,\infty]$) can be proved using considerably less complex techniques and without imposing assumptions on boundedness or on uncorrelation of the kernels. If, in addition, the variables $(\phi_m(x))$ are centered and uncorrelated, then the spectra are related as follows: $\operatorname{spec}(J) = \bigcup_{m=1}^{M} \operatorname{spec}(J_m)$; that is, $\{\lambda_i, i \geq 1\} = \bigcup_{m=1}^{M} \left\{ \lambda_i^{(m)}, i \geq 1 \right\}$. In that case one can equivalently write the bound of Theorem 4.6 as $R_r(H_p) \leq \sqrt{\frac{2}{n} \sum_{m=1}^{M} \sum_{j=1}^{\infty} \min(r, D^2 M^{\frac{2}{p^*}-1} \lambda_j^{(m)})} = \sqrt{\frac{2}{n} \left\| \left(\sum_{j=1}^{\infty} \min(r, D^2 M^{\frac{2}{p^*}-1} \lambda_j^{(m)}) \right)_{m=1}^{M} \right\|_1}$. However, the main focus of this thesis is on the more challenging case $1 \leq p \leq 2$, which is more relevant in practice (see empirical chapter of this thesis, i.e., Chapter 5).

Remark 3. It is interesting to compare the above bounds for the special case $p = 2$ with the ones of Bartlett et al. (2005). The main term of the bound of Theorem 4.6 (taking $t = p = 2$) is then essentially determined by $O\left(\sqrt{\frac{1}{n} \sum_{m=1}^{M} \sum_{j=1}^{\infty} \min\left(r, \lambda_j^{(m)}\right)} \right)$. If the variables $(\phi_m(x))$ are centered and uncorrelated, by the relation between the spectra stated in Remark 2, this is equivalently of order $O\left(\sqrt{\frac{1}{n} \sum_{j=1}^{\infty} \min\left(r, \lambda_j\right)} \right)$, which is also what we obtain through Theorem 4.6, and which coincides with the rate shown in Bartlett et al. (2005).

Proof of Theorem 4.5. The proof is based on first relating the complexity of the class H_p to its centered counterpart, i.e., where all functions $f_{\boldsymbol{w}} \in H_p$ are centered around their expected value. Then we compute the complexity of the centered class by decomposing the complexity into blocks, applying the no-correlation assumption, and using the inequalities of Hölder and Rosenthal. Then we relate it back to the original class, which, in the final step, we relate to a bound involving the truncation of the particular spectra of the kernels. Note that it suffices to prove the result for $t = p$ as trivially $\|\boldsymbol{w}\|_{2,t} \leq \|\boldsymbol{w}\|_{2,p}$ holds for all $t \geq p$ so that $H_p \subseteq H_t$ and therefore $R_r(H_p) \leq R_r(H_t)$.

Step 1: Relating the original class with the centered class. In order to exploit the no-correlation assumption, we will work in large parts of the proof with the centered class $\widetilde{H_p} = \left\{\widetilde{f}_{\boldsymbol{w}} \;\middle|\; \|\boldsymbol{w}\|_{2,p} \leq D\right\}$, wherein $\widetilde{f}_{\boldsymbol{w}} : x \mapsto \langle \boldsymbol{w}, \widetilde{\phi}(x)\rangle$, and $\widetilde{\phi}(x) := \phi(x) - \mathbb{E}\phi(x)$. We start the proof by noting that $\widetilde{f}_{\boldsymbol{w}}(x) = f_{\boldsymbol{w}}(x) - \langle \boldsymbol{w}, \mathbb{E}\phi(x)\rangle = f_{\boldsymbol{w}}(x) - \mathbb{E}\langle \boldsymbol{w}, \phi(x)\rangle = f_{\boldsymbol{w}}(\phi(x)) - \mathbb{E}f_{\boldsymbol{w}}(\phi(x))$, so that, by the bias-variance decomposition, it holds that

$$Pf_{\boldsymbol{w}}^2 = \mathbb{E}f_{\boldsymbol{w}}(x)^2 = \mathbb{E}\left(f_{\boldsymbol{w}}(x) - \mathbb{E}f_w(x)\right)^2 + \left(\mathbb{E}f_w(\phi(x))\right)^2 = P\widetilde{f}_{\boldsymbol{w}}^2 \;+\; \left(Pf_{\boldsymbol{w}}\right)^2. \tag{4.3}$$

Furthermore, we note that by Jensen's inequality

$$\begin{aligned}\left\|\mathbb{E}\phi(x)\right\|_{2,p^*} = \;& \left(\sum_{m=1}^{M} \left\|\mathbb{E}\phi_m(x)\right\|_2^{p^*}\right)^{\frac{1}{p^*}} = \left(\sum_{m=1}^{M} \left\langle \mathbb{E}\phi_m(x), \mathbb{E}\phi_m(x)\right\rangle^{\frac{p^*}{2}}\right)^{\frac{1}{p^*}} \\ \overset{\text{Jensen}}{\leq} & \left(\sum_{m=1}^{M} \mathbb{E}\left\langle \phi_m(x), \phi_m(x)\right\rangle^{\frac{p^*}{2}}\right)^{\frac{1}{p^*}} = \sqrt{\left\|\left(\operatorname{tr}(J_m)\right)_{m=1}^{M}\right\|_{\frac{p^*}{2}}}\end{aligned} \tag{4.4}$$

so that we can express the complexity of the centered class in terms of the uncentered one as follows:

$$\begin{aligned}R_r(H_p) &= \mathbb{E}\sup_{\substack{f_{\boldsymbol{w}}\in H_p,\\ Pf_{\boldsymbol{w}}^2 \leq r}} \left\langle \boldsymbol{w}, \frac{1}{n}\sum_{i=1}^{n}\sigma_i\phi(x_i)\right\rangle \\ &\leq \mathbb{E}\sup_{\substack{f_{\boldsymbol{w}}\in H_p,\\ Pf_{\boldsymbol{w}}^2 \leq r}} \left\langle \boldsymbol{w}, \frac{1}{n}\sum_{i=1}^{n}\sigma_i\widetilde{\phi}(x_i)\right\rangle + \mathbb{E}\sup_{\substack{f_{\boldsymbol{w}}\in H_p,\\ Pf_{\boldsymbol{w}}^2 \leq r}} \left\langle \boldsymbol{w}, \frac{1}{n}\sum_{i=1}^{n}\sigma_i\mathbb{E}\phi(x)\right\rangle.\end{aligned}$$

Concerning the first term of the above upper bound, using (4.3) we have $P\widetilde{f}_{\boldsymbol{w}}^2 \leq Pf_{\boldsymbol{w}}^2$, and thus

$$\mathbb{E}\sup_{\substack{f_{\boldsymbol{w}}\in H_p,\\ Pf_{\boldsymbol{w}}^2 \leq r}} \left\langle \boldsymbol{w}, \frac{1}{n}\sum_{i=1}^{n}\sigma_i\widetilde{\phi}(x_i)\right\rangle \leq \mathbb{E}\sup_{\substack{f_{\boldsymbol{w}}\in H_p,\\ P\widetilde{f}_{\boldsymbol{w}}^2 \leq r}} \left\langle \boldsymbol{w}, \frac{1}{n}\sum_{i=1}^{n}\sigma_i\widetilde{\phi}(x_i)\right\rangle = R_r(\widetilde{H}_p).$$

Now to bound the second term, we write

$$\begin{aligned}\mathbb{E}\sup_{\substack{f_{\boldsymbol{w}}\in H_p,\\ Pf_{\boldsymbol{w}}^2\le r}}\Big\langle \boldsymbol{w}, \frac{1}{n}\sum_{i=1}^n \sigma_i \mathbb{E}\phi(x)\Big\rangle &= \mathbb{E}\left|\frac{1}{n}\sum_{i=1}^n \sigma_i\right| \sup_{\substack{f_{\boldsymbol{w}}\in H_p,\\ Pf_{\boldsymbol{w}}^2\le r}} \langle \boldsymbol{w}, \mathbb{E}\phi(x)\rangle \\ &\le \sup_{\substack{f_{\boldsymbol{w}}\in H_p,\\ Pf_{\boldsymbol{w}}^2\le r}} \big\langle \boldsymbol{w}, \mathbb{E}\phi(x)\big\rangle \left(\mathbb{E}\left(\frac{1}{n}\sum_{i=1}^n \sigma_i\right)^2\right)^{\frac{1}{2}} \\ &= \sqrt{n} \sup_{\substack{f_{\boldsymbol{w}}\in H_p,\\ Pf_{\boldsymbol{w}}^2\le r}} \big\langle \boldsymbol{w}, \mathbb{E}\phi(x)\big\rangle\,.\end{aligned}$$

We finally observe

$$\big\langle \boldsymbol{w}, \mathbb{E}\phi(x)\big\rangle \overset{\text{Hölder}}{\le} \|\boldsymbol{w}\|_{2,p}\, \|\mathbb{E}\phi(x)\|_{2,p^*} \overset{(4.4)}{\le} \|\boldsymbol{w}\|_{2,p}\, \sqrt{\Big\|\big(\operatorname{tr}(J_m)\big)_{m=1}^M\Big\|_{\frac{p^*}{2}}}$$

as well as

$$\big\langle \boldsymbol{w}, \mathbb{E}\phi(x)\big\rangle = \mathbb{E} f_{\boldsymbol{w}}(x) \le \sqrt{P f_{\boldsymbol{w}}^2}\,.$$

Putting the above steps together, we obtain as a result

$$R_r(H_p) \le R_r(\widetilde{H}_p) + n^{-\frac{1}{2}} \min\left(\sqrt{r}, D\sqrt{\Big\|\big(\operatorname{tr}(J_m)\big)_{m=1}^M\Big\|_{\frac{p^*}{2}}}\right). \tag{4.5}$$

This shows that, at the expense of the additional summand on the right-hand side, we can work with the centered class instead of the uncentered one.

Step 2: Bounding the complexity of the centered class. Since the (centered) covariance operator $\mathbb{E}\widetilde{\phi}_m(x)\otimes\widetilde{\phi}_m(x)$ is also a self-adjoint Hilbert-Schmidt operator on $\mathcal{H}_m$, there exists an eigendecomposition

$$\mathbb{E}\widetilde{\phi}_m(x)\otimes\widetilde{\phi}_m(x) = \sum_{j=1}^\infty \widetilde{\lambda}_j^{(m)} \widetilde{\boldsymbol{u}}_j^{(m)} \otimes \widetilde{\boldsymbol{u}}_j^{(m)}, \tag{4.6}$$

wherein $(\widetilde{\boldsymbol{u}}_j^{(m)})_{j\ge1}$ is an orthogonal basis of $\mathcal{H}_m$. Furthermore, the no-correlation assumption (U) entails $\mathbb{E}\widetilde{\phi}_l(x)\otimes\widetilde{\phi}_m(x) = \mathbf{0}$ for all $l\ne m$. As a consequence,

$$\begin{aligned}P\widetilde{f}_{\boldsymbol{w}}^2 &= \mathbb{E}(\widetilde{f}_{\boldsymbol{w}}(x))^2 = \mathbb{E}\Big(\sum_{m=1}^M \big\langle \boldsymbol{w}_m, \widetilde{\phi}_m(x)\big\rangle\Big)^2 = \sum_{l,m=1}^M \Big\langle \boldsymbol{w}_l, \big(\mathbb{E}\widetilde{\phi}_l(x)\otimes\widetilde{\phi}_m(x)\big)\boldsymbol{w}_m\Big\rangle \\ &\overset{(\mathrm{U})}{=} \sum_{m=1}^M \Big\langle \boldsymbol{w}_m, \big(\mathbb{E}\widetilde{\phi}_m(x)\otimes\widetilde{\phi}_m(x)\big)\boldsymbol{w}_m\Big\rangle = \sum_{m=1}^M\sum_{j=1}^\infty \widetilde{\lambda}_j^{(m)} \Big\langle \boldsymbol{w}_m, \widetilde{\boldsymbol{u}}_j^{(m)}\Big\rangle^2 \end{aligned}\tag{4.7}$$

and, for all j and m,

$$\begin{aligned}
\mathbb{E}\Big\langle \frac{1}{n}\sum_{i=1}^{n}\sigma_i\widetilde{\phi}_m(x_i), \widetilde{\boldsymbol{u}}_j^{(m)}\Big\rangle^2 &= \mathbb{E}\frac{1}{n^2}\sum_{i,l=1}^{n}\sigma_i\sigma_l\left\langle\widetilde{\phi}_m(x_i), \widetilde{\boldsymbol{u}}_j^{(m)}\right\rangle\left\langle\widetilde{\phi}_m(x_l), \widetilde{\boldsymbol{u}}_j^{(m)}\right\rangle \\
&\overset{\sigma \text{ i.i.d.}}{=} \mathbb{E}\frac{1}{n^2}\sum_{i=1}^{n}\left\langle\widetilde{\phi}_m(x_i), \widetilde{\boldsymbol{u}}_j^{(m)}\right\rangle^2 \\
&= \frac{1}{n}\Big\langle \widetilde{\boldsymbol{u}}_j^{(m)}, \Big(\underbrace{\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\,\widetilde{\phi}_m(x_i)\otimes\widetilde{\phi}_m(x_i)}_{=\mathbb{E}\widetilde{\phi}_m(x)\otimes\widetilde{\phi}_m(x)}\Big)\widetilde{\boldsymbol{u}}_j^{(m)}\Big\rangle \\
&= \frac{\widetilde{\lambda}_j^{(m)}}{n}. \qquad (4.8)
\end{aligned}$$

Let now $h_1, \ldots, h_M$ be arbitrary nonnegative integers. We can express the local Rademacher complexity in terms of the eigendecompositon (4.6) as follows

$$\begin{aligned}
&R_r(\widetilde{H}_p) \\
&= \mathbb{E}\sup_{f_{\boldsymbol{w}}\in\widetilde{H}_p: P\widetilde{f}_{\boldsymbol{w}}^2\le r}\Big\langle \boldsymbol{w}, \frac{1}{n}\sum_{i=1}^{n}\sigma_i\widetilde{\phi}(x_i)\Big\rangle \\
&= \mathbb{E}\sup_{f_{\boldsymbol{w}}\in\widetilde{H}_p: P\widetilde{f}_{\boldsymbol{w}}^2\le r}\Big\langle \big(\boldsymbol{w}^{(m)}\big)_{m=1}^{M}, \big(\frac{1}{n}\sum_{i=1}^{n}\sigma_i\widetilde{\phi}_m(x_i)\big)_{m=1}^{M}\Big\rangle \\
&\le \mathbb{E}\sup_{P\widetilde{f}_{\boldsymbol{w}}^2\le r}\Big\langle \Big(\sum_{j=1}^{h_m}\sqrt{\widetilde{\lambda}_j^{(m)}}\langle\boldsymbol{w}^{(m)}, \widetilde{\boldsymbol{u}}_j^{(m)}\rangle\widetilde{\boldsymbol{u}}_j^{(m)}\Big)_{m=1}^{M}, \\
&\qquad\qquad \Big(\sum_{j=1}^{h_m}\sqrt{\widetilde{\lambda}_j^{(m)}}^{-1}\langle\frac{1}{n}\sum_{i=1}^{n}\sigma_i\widetilde{\phi}_m(x_i), \widetilde{\boldsymbol{u}}_j^{(m)}\rangle\widetilde{\boldsymbol{u}}_j^{(m)}\Big)_{m=1}^{M}\Big\rangle \\
&\qquad + \mathbb{E}\sup_{f_{\boldsymbol{w}}\in\widetilde{H}_p}\Big\langle \boldsymbol{w}, \Big(\sum_{j=h_m+1}^{\infty}\langle\frac{1}{n}\sum_{i=1}^{n}\sigma_i\widetilde{\phi}_m(x_i), \widetilde{\boldsymbol{u}}_j^{(m)}\rangle\widetilde{\boldsymbol{u}}_j^{(m)}\Big)_{m=1}^{M}\Big\rangle \\
&\overset{\text{C.-S., Jensen}}{\le} \sup_{P\widetilde{f}_{\boldsymbol{w}}^2\le r}\Bigg[\Bigg(\sum_{m=1}^{M}\sum_{j=1}^{h_m}\widetilde{\lambda}_j^{(m)}\langle\boldsymbol{w}^{(m)}, \widetilde{\boldsymbol{u}}_j^{(m)}\rangle^2\Bigg)^{\frac{1}{2}} \\
&\qquad\qquad \times\Bigg(\sum_{m=1}^{M}\sum_{j=1}^{h_m}\Big(\widetilde{\lambda}_j^{(m)}\Big)^{-1}\mathbb{E}\langle\frac{1}{n}\sum_{i=1}^{n}\sigma_i\widetilde{\phi}_m(x_i), \widetilde{\boldsymbol{u}}_j^{(m)}\rangle^2\Bigg)^{\frac{1}{2}}\Bigg] \\
&\qquad + \mathbb{E}\sup_{f_{\boldsymbol{w}}\in\widetilde{H}_p}\Big\langle \boldsymbol{w}, \Big(\sum_{j=h_m+1}^{\infty}\langle\frac{1}{n}\sum_{i=1}^{n}\sigma_i\widetilde{\phi}_m(x_i), \widetilde{\boldsymbol{u}}_j^{(m)}\rangle\widetilde{\boldsymbol{u}}_j^{(m)}\Big)_{m=1}^{M}\Big\rangle
\end{aligned}$$

so that (4.7) and (4.8) yield

$$
\begin{aligned}
&R_r(\widetilde{H}_p)\\
&\overset{(4.7),\,(4.8)}{\leq} \sqrt{\frac{r\sum_{m=1}^M h_m}{n}} + \mathbb{E}\sup_{f_{\boldsymbol{w}}\in\widetilde{H}_p}\Big\langle \boldsymbol{w}, \Big(\sum_{j=h_m+1}^{\infty}\langle\frac{1}{n}\sum_{i=1}^n \sigma_i\widetilde{\phi}_m(x_i),\widetilde{\boldsymbol{u}}_j^{(m)}\rangle\widetilde{\boldsymbol{u}}_j^{(m)}\Big)_{m=1}^M\Big\rangle\\
&\overset{\text{Hölder}}{\leq} \sqrt{\frac{r\sum_{m=1}^M h_m}{n}} + D\,\mathbb{E}\Big\|\Big(\sum_{j=h_m+1}^{\infty}\langle\frac{1}{n}\sum_{i=1}^n \sigma_i\widetilde{\phi}_m(x_i),\widetilde{\boldsymbol{u}}_j^{(m)}\rangle\widetilde{\boldsymbol{u}}_j^{(m)}\Big)_{m=1}^M\Big\|_{2,p^*}.
\end{aligned}
\tag{4.9}
$$

Step 3: Khintchine-Kahane's and Rosenthal's inequalities. We can now use the Khintchine-Kahane (K.-K.) inequality (see Lemma 4.1 in Appendix C) to further bound the right term in the above expression as follows

$$
\begin{aligned}
&\mathbb{E}\Big\|\Big(\sum_{j=h_m+1}^{\infty}\langle\frac{1}{n}\sum_{i=1}^n \sigma_i\widetilde{\phi}_m(x_i),\widetilde{\boldsymbol{u}}_j^{(m)}\rangle\widetilde{\boldsymbol{u}}_j^{(m)}\Big)_{m=1}^M\Big\|_{2,p^*}\\
&\qquad\overset{\text{Jensen}}{\leq} \mathbb{E}\Big(\sum_{m=1}^M \mathbb{E}_{\boldsymbol{\sigma}}\Big\|\sum_{j=h_m+1}^{\infty}\langle\frac{1}{n}\sum_{i=1}^n \sigma_i\widetilde{\phi}_m(x_i),\widetilde{\boldsymbol{u}}_j^{(m)}\rangle\widetilde{\boldsymbol{u}}_j^{(m)}\Big\|_{\mathcal{H}_m}^{p^*}\Big)^{\frac{1}{p^*}}\\
&\qquad\overset{\text{K.-K.}}{\leq} \sqrt{\frac{p^*}{n}}\,\mathbb{E}\Big(\sum_{m=1}^M\Big(\sum_{j=h_m+1}^{\infty}\frac{1}{n}\sum_{i=1}^n\langle\widetilde{\phi}_m(x_i),\widetilde{\boldsymbol{u}}_j^{(m)}\rangle^2\Big)^{\frac{p^*}{2}}\Big)^{\frac{1}{p^*}}\\
&\qquad\overset{\text{Jensen}}{\leq} \sqrt{\frac{p^*}{n}}\,\Big(\sum_{m=1}^M\mathbb{E}\Big(\sum_{j=h_m+1}^{\infty}\frac{1}{n}\sum_{i=1}^n\langle\widetilde{\phi}_m(x_i),\widetilde{\boldsymbol{u}}_j^{(m)}\rangle^2\Big)^{\frac{p^*}{2}}\Big)^{\frac{1}{p^*}}.
\end{aligned}
$$

Note that for $p \geq 2$ it holds that $p^*/2 \leq 1$, and thus it suffices to employ Jensen's inequality once again in order to move the expectation operator inside the inner term. In the general case, we need a handle on the $\frac{p^*}{2}$-th moments and to this end employ

Lemma 4.3 (Rosenthal + Young), which yields

$$\begin{aligned}
&\Big(\sum_{m=1}^{M}\mathbb{E}\Big(\sum_{j=h_m+1}^{\infty}\frac{1}{n}\sum_{i=1}^{n}\langle\widetilde{\phi}_m(x_i),\widetilde{\boldsymbol{u}}_j^{(m)}\rangle^2\Big)^{\frac{p^*}{2}}\Big)^{\frac{1}{p^*}}\\
&\qquad\overset{\text{R+Y}}{\leq}\left(\sum_{m=1}^{M}(ep^*)^{\frac{p^*}{2}}\left(\Big(\frac{B}{n}\Big)^{\frac{p^*}{2}}+\Big(\sum_{j=h_m+1}^{\infty}\underbrace{\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\langle\widetilde{\phi}_m(x_i),\widetilde{\boldsymbol{u}}_j^{(m)}\rangle^2}_{=\widetilde{\lambda}_j^{(m)}}\Big)^{\frac{p^*}{2}}\right)\right)^{\frac{1}{p^*}}\\
&\qquad\overset{(*)}{\leq}\sqrt{ep^*\left(\frac{BM^{\frac{2}{p^*}}}{n}+\Big(\sum_{m=1}^{M}\Big(\sum_{j=h_m+1}^{\infty}\widetilde{\lambda}_j^{(m)}\Big)^{\frac{p^*}{2}}\Big)^{\frac{2}{p^*}}\right)}\\
&\qquad=\sqrt{ep^*\left(\frac{BM^{\frac{2}{p^*}}}{n}+\Big\|\Big(\sum_{j=h_m+1}^{\infty}\widetilde{\lambda}_j^{(m)}\Big)_{m=1}^{M}\Big\|_{\frac{p^*}{2}}\right)}\\
&\qquad\leq\sqrt{ep^*\left(\frac{BM^{\frac{2}{p^*}}}{n}+\Big\|\Big(\sum_{j=h_m+1}^{\infty}\lambda_j^{(m)}\Big)_{m=1}^{M}\Big\|_{\frac{p^*}{2}}\right)}
\end{aligned}$$

where for $(*)$ we used the subadditivity of $\sqrt[p^*]{\cdot}$ and in the last step we exploit the eigenvalues of the centered covariance operator being smaller than or equal to those of the centered one: $\forall j,m:\ \widetilde{\lambda}_j^{(m)}\leq\lambda_j^{(m)}$. This follows from the decomposition $\mathbb{E}\phi_m(x)\otimes\phi_m(x)=\mathbb{E}\widetilde{\phi}_m(x)\otimes\widetilde{\phi}_m(x)+\mathbb{E}\phi_m(x)\otimes\mathbb{E}\phi_m(x)$ by the Lidskii-Mirsky-Wielandt theorem. Thus putting the pieces together, we obtain from (4.9)

$$\begin{aligned}
&R_r(\widetilde{H}_p)\\
&\quad\leq\ \sqrt{\frac{r\sum_{m=1}^{M}h_m}{n}}+D\sqrt{\frac{ep^{*2}}{n}\left(\frac{BM^{\frac{2}{p^*}}}{n}+\Big\|\Big(\sum_{j=h_m+1}^{\infty}\lambda_j^{(m)}\Big)_{m=1}^{M}\Big\|_{\frac{p^*}{2}}\right)}\\
&\quad=\ \sqrt{\frac{r\sum_{m=1}^{M}h_m}{n}}+\sqrt{\frac{ep^{*2}D^2}{n}\Big\|\Big(\sum_{j=h_m+1}^{\infty}\lambda_j^{(m)}\Big)_{m=1}^{M}\Big\|_{\frac{p^*}{2}}}+\frac{\sqrt{Be}DM^{\frac{1}{p^*}}p^*}{n},\quad(4.10)
\end{aligned}$$

again, using subadditivity for the last inequality.

Step 4: Bounding the complexity of the original class. Now note that for all nonnegative integers h_m we either have

$$n^{-\frac{1}{2}}\min\Big(\sqrt{r},D\sqrt{\big\|\big(\mathrm{tr}(J_m)\big)_{m=1}^{M}\big\|_{\frac{p^*}{2}}}\Big)\leq\sqrt{\frac{ep^{*2}D^2}{n}\Big\|\Big(\sum_{j=h_m+1}^{\infty}\lambda_j^{(m)}\Big)_{m=1}^{M}\Big\|_{\frac{p^*}{2}}}$$

(in case all h_m are zero) or it holds

$$n^{-\frac{1}{2}}\min\Big(\sqrt{r},D\sqrt{\big\|\big(\mathrm{tr}(J_m)\big)_{m=1}^{M}\big\|_{\frac{p^*}{2}}}\Big)\leq\sqrt{\frac{r\sum_{m=1}^{M}h_m}{n}}$$

(in case that at least one h_m is nonzero) so that in any case we get

$$n^{-\frac{1}{2}} \min\left(\sqrt{r}, D\sqrt{\left\|\left(\operatorname{tr}(J_m)\right)_{m=1}^M\right\|_{\frac{p^*}{2}}}\right) \leq \sqrt{\frac{r\sum_{m=1}^M h_m}{n}} + \sqrt{\frac{ep^{*2}D^2}{n}\left\|\left(\sum_{j=h_m+1}^{\infty}\lambda_j^{(m)}\right)_{m=1}^M\right\|_{\frac{p^*}{2}}}. \quad (4.11)$$

Thus the following preliminary bound follows from (4.5) by (4.10) and (4.11):

$$R_r(H_p) \leq \sqrt{\frac{4r\sum_{m=1}^M h_m}{n}} + \sqrt{\frac{4ep^{*2}D^2}{n}\left\|\left(\sum_{j=h_m+1}^{\infty}\lambda_j^{(m)}\right)_{m=1}^M\right\|_{\frac{p^*}{2}}} + \frac{\sqrt{Be}DM^{\frac{1}{p^*}}p^*}{n}, \quad (4.12)$$

for all nonnegative integers $h_m \geq 0$. We could stop here as the above bound is already the one that will be used in the subsequent section for the computation of the excess loss bounds. However, we can work a little more on the form of the above bound to gain more insight into the properties—we will show that it is related to the truncation of the spectra at the scale r.

Step 5: Relating the bound to the truncation of the spectra of the kernels. To this end, notice that for all nonnegative real numbers A_1, A_2 and any $\boldsymbol{a}_1, \boldsymbol{a}_2 \in \mathbb{R}_+^m$ it holds for all $q \geq 1$

$$\sqrt{A_1} + \sqrt{A_2} \leq \sqrt{2(A_1+A_2)} \quad (4.13)$$

$$\|\boldsymbol{a}_1\|_q + \|\boldsymbol{a}_2\|_q \leq 2^{1-\frac{1}{q}}\|\boldsymbol{a}_1+\boldsymbol{a}_2\|_q \leq 2\|\boldsymbol{a}_1+\boldsymbol{a}_2\|_q \quad (4.14)$$

(the first statement follows from the concavity of the square root function and the second one is proved in appendix C; see Lemma C.3) and thus

$$\begin{aligned} R_r(H_p) &\overset{(4.13)}{\leq} \sqrt{8\left(\frac{r\sum_{m=1}^M h_m}{n} + \frac{ep^{*2}D^2}{n}\left\|\left(\sum_{j=h_m+1}^{\infty}\lambda_j^{(m)}\right)_{m=1}^M\right\|_{\frac{p^*}{2}}\right)} + \frac{\sqrt{Be}DM^{\frac{1}{p^*}}p^*}{n} \\ &\overset{\ell_1\text{-to-}\ell_{\frac{p^*}{2}}}{\leq} \sqrt{\frac{8}{n}\left(rM^{1-\frac{2}{p^*}}\left\|\left(h_m\right)_{m=1}^M\right\|_{\frac{p^*}{2}} + ep^{*2}D^2\left\|\left(\sum_{j=h_m+1}^{\infty}\lambda_j^{(m)}\right)_{m=1}^M\right\|_{\frac{p^*}{2}}\right)} + \frac{\sqrt{Be}DM^{\frac{1}{p^*}}p^*}{n} \\ &\overset{(4.14)}{\leq} \sqrt{\frac{16}{n}\left\|\left(rM^{1-\frac{2}{p^*}}h_m + ep^{*2}D^2\sum_{j=h_m+1}^{\infty}\lambda_j^{(m)}\right)_{m=1}^M\right\|_{\frac{p^*}{2}}} + \frac{\sqrt{Be}DM^{\frac{1}{p^*}}p^*}{n}, \end{aligned}$$

where, in order to obtain the second inequality, we apply that for all non-negative $\boldsymbol{a} \in \mathbb{R}^M$ and $0 < q < p \leq \infty$ it holds[6]

$$(\ell_q\text{-to-}\ell_p \text{ conversion}) \qquad \|\boldsymbol{a}\|_q = \langle \mathbf{1}, \boldsymbol{a}^q \rangle^{\frac{1}{q}} \overset{\text{Hölder}}{\leq} \left(\|\mathbf{1}\|_{(p/q)^*} \|\boldsymbol{a}^q\|_{p/q} \right)^{1/q} = M^{\frac{1}{q}-\frac{1}{p}} \|\boldsymbol{a}\|_p . \tag{4.15}$$

Since the above holds for all nonnegative integers h_m, it follows

$$\begin{aligned} R_r(H_p) &\leq \sqrt{\frac{16}{n} \left\| \left(\min_{h_m \geq 0} \; r M^{1-\frac{2}{p^*}} h_m + e p^{*2} D^2 \sum_{j=h_m+1}^{\infty} \lambda_j^{(m)} \right)_{m=1}^{M} \right\|_{\frac{p^*}{2}}} + \frac{\sqrt{Be} D M^{\frac{1}{p^*}} p^*}{n} \\ &= \sqrt{\frac{16}{n} \left\| \left(\sum_{j=1}^{\infty} \min \left(r M^{1-\frac{2}{p^*}}, e p^{*2} D^2 \lambda_j^{(m)} \right) \right)_{m=1}^{M} \right\|_{\frac{p^*}{2}}} + \frac{\sqrt{Be} D M^{\frac{1}{p^*}} p^*}{n}, \end{aligned}$$

which completes the proof of the theorem. ■

Proof of Remark 1. To see that Remark 1 holds, notice that $R(H_{p=1}) \leq R(H_t)$ for all $t \geq 1$ and thus, by choosing $t = (\log(M))^*$ in the bound of Theorem 4.5 (which is valid only if $t \in [1, 2]$, i.e., $M \geq e^2$), we obtain

$$\begin{aligned} R_r(H_1) &\leq \sqrt{\frac{16}{n} \left\| \left(\sum_{j=1}^{\infty} \min \left(r M^{1-\frac{2}{t^*}}, e t^{*2} D^2 \lambda_j^{(m)} \right) \right)_{m=1}^{M} \right\|_{\frac{t^*}{2}}} + \frac{\sqrt{Be} D M^{\frac{1}{t^*}} t^*}{n} \\ &\overset{\ell_{\frac{t^*}{2}}\text{-to-}\ell_\infty}{\leq} \sqrt{\frac{16}{n} \left\| \left(\sum_{j=1}^{\infty} \min \left(r M, e t^{*2} M^{\frac{2}{t^*}} D^2 \lambda_j^{(m)} \right) \right)_{m=1}^{M} \right\|_{\infty}} + \frac{\sqrt{Be} D M^{\frac{1}{t^*}} t^*}{n} \\ &= \sqrt{\frac{16}{n} \left\| \left(\sum_{j=1}^{\infty} \min \left(r M, e^3 D^2 (\log M)^2 \lambda_j^{(m)} \right) \right)_{m=1}^{M} \right\|_{\infty}} + \frac{\sqrt{B} e^{\frac{3}{2}} D (\log M)}{n}, \end{aligned}$$

which completes the proof. ■

Proof of Theorem 4.6. The eigendecomposition $\mathbb{E}\phi(x) \otimes \phi(x) = \sum_{j=1}^{\infty} \lambda_j \boldsymbol{u}_j \otimes \boldsymbol{u}_j$ yields

$$P f_{\boldsymbol{w}}^2 = \mathbb{E}(f_{\boldsymbol{w}}(x))^2 = \mathbb{E}\langle \boldsymbol{w}, \phi(x) \rangle^2 = \left\langle \boldsymbol{w}, (\mathbb{E}\phi(x) \otimes \phi(x)) \boldsymbol{w} \right\rangle = \sum_{j=1}^{\infty} \lambda_j \langle \boldsymbol{w}, \boldsymbol{u}_j \rangle^2 , \tag{4.16}$$

[6] We denote by $\boldsymbol{a}^q$ the vector with entries a_i^q and by $\mathbf{1}$ the vector with entries all 1.

and, for all j

$$\begin{aligned}\mathbb{E}\Big\langle \frac{1}{n}\sum_{i=1}^n \sigma_i\phi(x_i), \boldsymbol{u}_j\Big\rangle^2 &= \mathbb{E}\frac{1}{n^2}\sum_{i,l=1}^n \sigma_i\sigma_l \left\langle\phi(x_i),\boldsymbol{u}_j\right\rangle\left\langle\phi(x_l),\boldsymbol{u}_j\right\rangle \overset{\sigma \text{ i.i.d.}}{=} \mathbb{E}\frac{1}{n^2}\sum_{i=1}^n \left\langle\phi(x_i),\boldsymbol{u}_j\right\rangle^2 \\ &= \frac{1}{n}\Big\langle \boldsymbol{u}_j, \Big(\underbrace{\frac{1}{n}\sum_{i=1}^n \mathbb{E}\phi(x_i)\otimes\phi(x_i)}_{=\mathbb{E}\phi(x)\otimes\phi(x)}\Big)\boldsymbol{u}_j\Big\rangle = \frac{\lambda_j}{n}. \end{aligned} \tag{4.17}$$

Therefore, we can use, for any nonnegative integer h, the Cauchy-Schwarz inequality and a block-structured version of Hölder's inequality (see Lemma C.1) to bound the local Rademacher complexity as follows:

$$\begin{aligned} &R_r(H_p) \\ &= \mathbb{E}\sup_{f_{\boldsymbol{w}}\in H_p: Pf_{\boldsymbol{w}}^2\le r}\big\langle \boldsymbol{w}, \frac{1}{n}\sum_{i=1}^n \sigma_i\phi(x_i)\big\rangle \\ &= \mathbb{E}\sup_{f_{\boldsymbol{w}}\in H_p: Pf_{\boldsymbol{w}}^2\le r}\Big\langle \sum_{j=1}^h \sqrt{\lambda_j}\langle \boldsymbol{w},\boldsymbol{u}_j\rangle\boldsymbol{u}_j, \sum_{j=1}^h \sqrt{\lambda_j}^{-1}\langle\frac{1}{n}\sum_{i=1}^n\sigma_i\phi(x_i),\boldsymbol{u}_j\rangle\boldsymbol{u}_j\Big\rangle \\ &\qquad + \Big\langle \boldsymbol{w}, \sum_{j=h+1}^\infty \langle\frac{1}{n}\sum_{i=1}^n\sigma_i\phi(x_i),\boldsymbol{u}_j\rangle\boldsymbol{u}_j\Big\rangle \\ &\overset{\text{C.-S., (4.16), (4.17)}}{\le} \sqrt{\frac{rh}{n}} + \mathbb{E}\sup_{f_{\boldsymbol{w}}\in H_p}\Big\langle \boldsymbol{w}, \sum_{j=h+1}^\infty \langle\frac{1}{n}\sum_{i=1}^n\sigma_i\phi(x_i),\boldsymbol{u}_j\rangle\boldsymbol{u}_j\Big\rangle \\ &\overset{\text{Hölder}}{\le} \sqrt{\frac{rh}{n}} + D\mathbb{E}\Big\| \sum_{j=h+1}^\infty \langle\frac{1}{n}\sum_{i=1}^n\sigma_i\phi(x_i),\boldsymbol{u}_j\rangle\boldsymbol{u}_j\Big\|_{2,p^*} \\ &\overset{\ell_{\frac{p^*}{2}}\text{-to-}\ell_2}{\le} \sqrt{\frac{rh}{n}} + DM^{\frac{1}{p^*}-\frac{1}{2}}\mathbb{E}\Big\| \sum_{j=h+1}^\infty \langle\frac{1}{n}\sum_{i=1}^n\sigma_i\phi(x_i),\boldsymbol{u}_j\rangle\boldsymbol{u}_j\Big\|_2 \\ &\overset{\text{Jensen}}{\le} \sqrt{\frac{rh}{n}} + DM^{\frac{1}{p^*}-\frac{1}{2}}\Big(\sum_{j=h+1}^\infty \underbrace{\mathbb{E}\langle\frac{1}{n}\sum_{i=1}^n\sigma_i\phi(x_i),\boldsymbol{u}_j\rangle^2}_{\overset{(4.17)}{\le}\frac{\lambda_j}{n}}\Big)^{\frac{1}{2}} \\ &\le \sqrt{\frac{rh}{n}} + \sqrt{\frac{D^2M^{\frac{2}{p^*}-1}}{n}\sum_{j=h+1}^\infty \lambda_j}. \end{aligned}$$

Since the above holds for all h, the result now follows from $\sqrt{A}+\sqrt{B}\le\sqrt{2(A+B)}$ for all nonnegative real numbers A, B (which holds by the concavity of the square root

function):

$$R_r(H_p) \leq \sqrt{\frac{2}{n} \min_{0 \leq h \leq n} \left(rh + D^2 M^{\frac{2}{p^*}-1} \sum_{j=h+1}^{\infty} \lambda_j \right)} = \sqrt{\frac{2}{n} \sum_{j=1}^{\infty} \min(r, D^2 M^{\frac{2}{p^*}-1} \lambda_j)} \,.$$

■

4.2.1 Lower Bound

In this subsection we investigate the tightness of our bound on the local Rademacher complexity of H_p. To derive a lower bound, we consider the particular case where variables $\phi_1(x), \ldots, \phi_M(x)$ are i.i.d. For example, this happens if the original input space $\mathcal{X}$ is $\mathbb{R}^M$, the original input variable $x \in \mathcal{X}$ has i.i.d. coordinates, and the kernels $k_1, \ldots, k_M$ are equal, but each acts on a different coordinate of x.

Lemma 4.7. *Assume that the variables $\phi_1(x), \ldots, \phi_M(x)$ are centered and identically independently distributed. Then, the following lower bound holds for the local Rademacher complexity of H_p for any $p \geq 1$:*

$$R_r(H_{p,D,M}) \;\geq\; R_{rM}(H_{1,DM^{1/p^*},1}).$$

Proof First note that, since the $x^{(i)}$ are centered and uncorrelated,

$$Pf_{\boldsymbol{w}}^2 = \Big(\sum_{m=1}^{M} \big\langle \boldsymbol{w}_m, \phi_m(x) \big\rangle \Big)^2 = \sum_{m=1}^{M} \big\langle \boldsymbol{w}_m, \phi_m(x) \big\rangle^2.$$

Now it follows

$$\begin{aligned}
R_r(H_{p,D,M}) &= \mathbb{E} \sup_{\substack{\boldsymbol{w}: \\ \substack{Pf_{\boldsymbol{w}}^2 \le r \\ \|\boldsymbol{w}\|_{2,p} \le D}}} \left\langle \boldsymbol{w}, \frac{1}{n}\sum_{i=1}^n \sigma_i \phi(x_i) \right\rangle \\
&= \mathbb{E} \sup_{\substack{\boldsymbol{w}: \\ \substack{\sum_{m=1}^M \left\langle \boldsymbol{w}^{(m)}, \phi_m(x) \right\rangle^2 \le r \\ \|\boldsymbol{w}\|_{2,p} \le D}}} \left\langle w, \frac{1}{n}\sum_{i=1}^n \sigma_i \phi(x_i) \right\rangle \\
&\ge \mathbb{E} \sup_{\substack{\boldsymbol{w}: \\ \substack{\forall m\colon \left\langle \boldsymbol{w}^{(m)}, \phi_m(x) \right\rangle^2 \le r/M \\ \left\|\boldsymbol{w}^{(m)}\right\|_{2,p} \le D \\ \left\|\boldsymbol{w}^{(1)}\right\| = \cdots = \left\|\boldsymbol{w}^{(M)}\right\|}}} \left\langle \boldsymbol{w}, \frac{1}{n}\sum_{i=1}^n \sigma_i \phi(x_i) \right\rangle \\
&= \mathbb{E} \sup_{\substack{\boldsymbol{w}: \\ \substack{\forall m\colon \left\langle \boldsymbol{w}^{(m)}, \phi_m(x) \right\rangle^2 \le r/M \\ \forall m\colon \left\|\boldsymbol{w}^{(m)}\right\|_2 \le DM^{-\frac{1}{p}}}}} \sum_{m=1}^M \left\langle \boldsymbol{w}^{(m)}, \frac{1}{n}\sum_{i=1}^n \sigma_i \phi_m(x_i) \right\rangle \\
&= \sum_{m=1}^M \mathbb{E} \sup_{\substack{\boldsymbol{w}^{(m)}: \\ \left\langle \boldsymbol{w}^{(m)}, \phi_m(x) \right\rangle^2 \le r/M \\ \left\|\boldsymbol{w}^{(m)}\right\|_2 \le DM^{-\frac{1}{p}}}} \left\langle \boldsymbol{w}^{(m)}, \frac{1}{n}\sum_{i=1}^n \sigma_i \phi_m(x_i) \right\rangle
\end{aligned}$$

so that we can use the i.i.d. assumption on $\phi_m(x)$ to equivalently rewrite the last term as

$$\begin{aligned}
R_r(H_{p,D,M}) &\overset{(\phi_m(x))_{1\le m\le M} \text{ i.i.d.}}{\ge} \mathbb{E} \sup_{\substack{\boldsymbol{w}^{(1)}: \\ \left\langle \boldsymbol{w}^{(1)}, \phi_1(x) \right\rangle^2 \le r/M \\ \left\|\boldsymbol{w}^{(1)}\right\|_2 \le DM^{-\frac{1}{p}}}} \left\langle M\boldsymbol{w}^{(1)}, \frac{1}{n}\sum_{i=1}^n \sigma_i \phi_1(x_i) \right\rangle \\
&= \mathbb{E} \sup_{\substack{\boldsymbol{w}^{(1)}: \\ \left\langle M\boldsymbol{w}^{(1)}, \phi_1(x) \right\rangle^2 \le rM \\ \left\|M\boldsymbol{w}^{(1)}\right\|_2 \le DM^{\frac{1}{p^*}}}} \left\langle M\boldsymbol{w}^{(1)}, \frac{1}{n}\sum_{i=1}^n \sigma_i \phi_1(x_i) \right\rangle \\
&= \mathbb{E} \sup_{\substack{\boldsymbol{w}^{(1)}: \\ \left\langle \boldsymbol{w}^{(1)}, \phi_1(x) \right\rangle^2 \le rM \\ \left\|\boldsymbol{w}^{(1)}\right\|_2 \le DM^{\frac{1}{p^*}}}} \left\langle \boldsymbol{w}^{(1)}, \frac{1}{n}\sum_{i=1}^n \sigma_i \phi_1(x_i) \right\rangle \\
&= R_{rM}(H_{1,DM^{1/p^*},1})\,.
\end{aligned}$$

■

In Mendelson (2003) it was shown that there is an absolute constant c so that if $\lambda^{(1)} \geq \frac{1}{n}$ then for all $r \geq \frac{1}{n}$ it holds $R_r(H_{1,1,1}) \geq \sqrt{\frac{c}{n}\sum_{j=1}^{\infty}\min(r, \lambda_j^{(1)})}$. Closer inspection of the proof reveals that more generally it holds $R_r(H_{1,D,1}) \geq \sqrt{\frac{c}{n}\sum_{j=1}^{\infty}\min(r, D^2\lambda_j^{(1)})}$ if $\lambda_1^{(m)} \geq \frac{1}{nD^2}$ so that we can use that result together with the previous lemma to obtain:

Theorem 4.8 (Lower bound)**.** *Assume that the kernels are centered and identically independently distributed. Then the following lower bound holds for the local Rademacher complexity of H_p. There is an absolute constant c such that if $\lambda^{(1)} \geq \frac{1}{nD^2}$ then for all $r \geq \frac{1}{n}$ and $p \geq 1$,*

$$R_r(H_{p,D,M}) \;\geq\; \sqrt{\frac{c}{n}\sum_{j=1}^{\infty}\min\Big(rM, D^2M^{2/p^*}\lambda_j^{(1)}\Big)}. \tag{4.18}$$

We would like to compare the above lower bound with the upper bound of Theorem 4.5. To this end, note that for centered identical independent kernels the upper bound reads

$$R_r(H_p) \leq \sqrt{\frac{16}{n}\sum_{j=1}^{\infty}\min\Big(rM, ceD^2p^{*2}M^{\frac{2}{p^*}}\lambda_j^{(1)}\Big)} + \frac{\sqrt{Be}DM^{\frac{1}{p^*}}p^*}{n},$$

which is of the order $O\big(\sqrt{\sum_{j=1}^{\infty}\min\big(rM, D^2M^{\frac{2}{p^*}}\lambda_j^{(1)}\big)}\big)$ and, disregarding the quickly converging term on the right hand side and absolute constants, again matches the upper bounds of the previous section. A similar comparison can be performed for the upper bound of Theorem 4.6: by Remark 2 the bound reads

$$R_r(H_p) \leq \sqrt{\frac{2}{n}\Big\|\Big(\sum_{j=1}^{\infty}\min\big(r, D^2M^{\frac{2}{p^*}-1}\lambda_j^{(m)}\big)\Big)_{m=1}^{M}\Big\|_1},$$

which for i.i.d. kernels becomes $\sqrt{2/n\sum_{j=1}^{\infty}\min\big(rM, D^2M^{\frac{2}{p^*}}\lambda_j^{(1)}\big)}$ and thus, up to constants, matches the lower bound. This shows that the upper bounds of the previous section are tight.

4.3 Excess Risk Bounds

In this section, we show an application of our results to prediction problems, such as classification or regression. To this aim, in addition to the data $x_1, \ldots, x_n$ introduced earlier in this thesis, let also a label sequence $y_1, \ldots, y_n \subset [-1, 1]$ be given so that the pairs $(x_1, y_1), \ldots, (x_n, y_n)$ are i.i.d. generated from a probability distribution. The goal in statistical learning is to find a hypothesis f from a given class $\mathcal{F}$ that minimizes the expected loss $\mathbb{E}\, l(f(x), y)$, where $l : \mathbb{R}^2 \mapsto [0, 1]$ is a predefined loss function that encodes the objective of the given learning/prediction task at hand. For example, the

hinge loss $l(t,y) = \max(0, 1-yt)$ and the squared loss $l(t,y) = (t-y)^2$ are frequently used in classification and regression problems, respectively.

Since the distribution generating the example/label pairs is unknown, the optimal decision function

$$f^* := \operatorname{argmin}_{f\in\mathcal{F}} \mathbb{E}\, l(f(x), y)$$

cannot be computed directly and a frequently used method consists in minimizing the *empirical* loss instead

$$\hat{f} := \operatorname{argmin}_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i).$$

In order to evaluate the performance of this so-called *empirical risk minimization* algorithm, we study the excess loss

$$P(l_{\hat{f}} - l_{f^*}) \;:=\; \mathbb{E}\, l(\hat{f}(x), y) - \mathbb{E}\, l(f^*(x), y).$$

In Bartlett et al. (2005) and Koltchinskii (2006) it was shown that the rate of convergence of the excess risk is basically determined by the fixed point of the local Rademacher complexity. For example, the following result is a slight modification of Corollary 5.3 in Bartlett et al. (2005) that is well-taylored to the class studied in this thesis.[7]

Lemma 4.9 (Bartlett et al., 2005)**.** *Let $\mathcal{F}$ be an absolute convex class ranging in the interval $[a,b]$ and let l be a Lipschitz continuous loss with constant L. Assume there is a positive constant F such that $\forall f \in \mathcal{F}:\ P(f-f^*)^2 \le F\, P(l_f - l_{f^*})$. Then, denoting by r^* the fixed point of*

$$2FL\, R_{\frac{r}{4L^2}}(\mathcal{F})$$

for all $z > 0$ with probability at least $1 - e^{-z}$ the excess loss can be bounded as

$$P(l_{\hat{f}} - l_{f^*}) \le 7\frac{r^*}{F} + \frac{(11L(b-a) + 27F)z}{n}.$$

The above result shows that in order to obtain an excess risk bound on the multi-kernel class H_p it suffices to compute the fixed point of our bound on the local Rademacher complexity presented in Section 4.2. To this end, we show:

Lemma 4.10. *Assume that $\|k\|_\infty \le B$ almost surely and let $p \in [1,2]$. For the fixed point r^* of the local Rademacher complexity $2FLR_{\frac{r}{4L^2}}(H_p)$ it holds*

$$r^* \le \min_{0\le h_m \le \infty} \frac{4F^2\sum_{m=1}^{M} h_m}{n} + 8FL\sqrt{\frac{ep^{*2}D^2}{n}\left\|\left(\sum_{j=h_m+1}^{\infty}\lambda_j^{(m)}\right)_{m=1}^{M}\right\|_{\frac{p^*}{2}}} + \frac{4\sqrt{Be}DFLM^{\frac{1}{p^*}}p^*}{n}.$$

[7] We exploit the improved constants from Theorem 3.3 in Bartlett et al. (2005) because an absolute convex class is star-shaped. Compared to Corollary 5.3 in Bartlett et al. (2005) we also use a slightly more general function class ranging in $[a,b]$ instead of the interval $[-1,1]$. This is justified by Theorem 3.3 as well.

Proof For this proof we make use of the bound (4.12) on the local Rademacher complexity. Defining $a = \frac{4F^2 \sum_{m=1}^{M} h_m}{n}$ and

$$b = 4FL \sqrt{\frac{ep^{*2}D^2}{n} \left\| \left(\sum_{j=h_m+1}^{\infty} \lambda_j^{(m)} \right)_{m=1}^{M} \right\|_{\frac{p^*}{2}}} + \frac{2\sqrt{Be}DFLM^{\frac{1}{p^*}}p^*}{n} ,$$

in order to find a fixed point of (4.12), we need to solve for $r = \sqrt{ar} + b$, which is equivalent to solving $r^2 - (a+2b)r + b^2 = 0$ for a positive root. Denote this solution by r^*. It is then easy to see that $r^* \geq a + 2b$. Re-substituting the definitions of a and b yields the result. ∎

We now address the issue of computing actual rates of convergence of the fixed point r^* under the assumption of algebraically decreasing eigenvalues of the kernel matrices, this means, we assume $\exists d_m : \ \lambda_j^{(m)} \leq d_m j^{-\alpha_m}$ for some $\alpha_m > 1$. This is a common assumption and, for example, fulfilled by finite rank kernels and convolution kernels (Williamson et al., 2001). Note that this implies

$$\begin{aligned} \sum_{j=h_m+1}^{\infty} \lambda_j^{(m)} &\leq d_m \sum_{j=h_m+1}^{\infty} j^{-\alpha_m} \leq d_m \int_{h_m}^{\infty} x^{-\alpha_m} dx = d_m \Big[\frac{1}{1-\alpha_m} x^{1-\alpha_m} \Big]_{h_m}^{\infty} \\ &= -\frac{d_m}{1-\alpha_m} h_m^{1-\alpha_m} . \end{aligned} \tag{4.19}$$

To exploit the above fact, first note that by ℓ_p-to-ℓ_q conversion

$$\frac{4F^2 \sum_{m=1}^{M} h_m}{n} \leq 4F \sqrt{\frac{F^2 M \sum_{m=1}^{M} h_m^2}{n^2}} \leq 4F \sqrt{\frac{F^2 M^{2-\frac{2}{p^*}} \left\| \left(h_m^2\right) \right)_{m=1}^{M} \right\|_{2/p^*}}{n^2}}$$

so that we can translate the result of the previous lemma by (4.13), (4.14), and (4.15) into

$$\begin{aligned} r^* \leq \min_{0 \leq h_m \leq \infty} 8F \sqrt{\frac{1}{n} \left\| \left(\frac{F^2 M^{2-\frac{2}{p^*}} h_m^2}{n} + 4ep^{*2}D^2L^2 \sum_{j=h_m+1}^{\infty} \lambda_j^{(m)} \right)_{m=1}^{M} \right\|_{\frac{p^*}{2}}} \\ + \frac{4\sqrt{Be}DFLM^{\frac{1}{p^*}}p^*}{n} . \end{aligned} \tag{4.20}$$

Inserting the result of (4.19) into the above bound and setting the derivative with respect to h_m to zero we find the optimal h_m as

$$h_m = \left(4d_m ep^{*2}D^2F^{-2}L^2M^{\frac{2}{p^*}-2}n \right)^{\frac{1}{1+\alpha_m}} .$$

Re-substituting the above into (4.20) we note that

$$r^* = O\left(\sqrt{\left\| \left(n^{-\frac{2\alpha_m}{1+\alpha_m}} \right)_{m=1}^{M} \right\|_{\frac{p^*}{2}}} \right)$$

so that we observe that the asymptotic rate of convergence in n is determined by the kernel with the smallest decreasing spectrum (i.e., smallest α_m).

Therefore, denoting $d_{\max} := \max_{m=1,\dots,M} d_m$, $\alpha_{\min} := \min_{m=1,\dots,M} \alpha_m$, and $h_{\max} := \left(4d_{\max}ep^{*2}D^2F^{-2}L^2M^{\frac{2}{p^*}-2}n\right)^{\frac{1}{1+\alpha_{\min}}}$, we can upper bound (4.20) by

$$\begin{aligned} r^* &\leq 8F\sqrt{\frac{3-\alpha_m}{1-\alpha_m}F^2M^2h_{\max}^2n^{-2}} + \frac{4\sqrt{Be}DFLM^{\frac{1}{p^*}}p^*}{n} \\ &\leq 8\sqrt{\frac{3-\alpha_m}{1-\alpha_m}}F^2Mh_{\max}n^{-1} + \frac{4\sqrt{Be}DFLM^{\frac{1}{p^*}}p^*}{n} \\ &\leq 16\sqrt{e\frac{3-\alpha_m}{1-\alpha_m}}(d_{\max}D^2L^2p^{*2})^{\frac{1}{1+\alpha_{\min}}}F^{\frac{2\alpha_{\min}}{1+\alpha_{\min}}}M^{1+\frac{2}{1+\alpha_{\min}}\left(\frac{1}{p^*}-1\right)}n^{-\frac{\alpha_{\min}}{1+\alpha_{\min}}} \\ &\quad + \frac{4\sqrt{Be}DFLM^{\frac{1}{p^*}}p^*}{n}\,. \end{aligned} \tag{4.21}$$

We have thus proved the following theorem, which follows by the above inequality, Lemma 4.9, and the fact that our class H_p ranges in $BDM^{\frac{1}{p^*}}$.

Theorem 4.11. *Assume that $\|k\|_\infty \leq B$ and $\exists d_{\max} > 0$ and $\alpha := \alpha_{\min} > 1$ such that for all $m = 1, \dots, M$ it holds $\lambda_j^{(m)} \leq d_{\max}j^{-\alpha}$. Let l be a Lipschitz continuous loss with constant L and assume there is a positive constant F such that $\forall f \in \mathcal{F}:\ P(f-f^*)^2 \leq F\,P(l_f - l_{f^*})$. Then for all $z > 0$ with probability at least $1-e^{-z}$ the excess loss of the multi-kernel class H_p can be bounded for $p \in [1, \dots, 2]$ as*

$$\begin{aligned} &P(l_{\hat{f}} - l_{f^*}) \\ &\quad\leq \min_{t\in[p,2]} \quad 186\sqrt{\frac{3-\alpha_m}{1-\alpha_m}}\left(d_{\max}D^2L^2t^{*2}\right)^{\frac{1}{1+\alpha}}F^{\frac{\alpha-1}{\alpha+1}}M^{1+\frac{2}{1+\alpha}\left(\frac{1}{t^*}-1\right)}n^{-\frac{\alpha}{1+\alpha}} \\ &\qquad\qquad + \frac{47\sqrt{B}DLM^{\frac{1}{t^*}}t^*}{n} + \frac{(22BDLM^{\frac{1}{t^*}}+27F)z}{n}\,. \end{aligned}$$

We see from the above bound that convergence can be almost as slow as $O\left(p^*M^{\frac{1}{p^*}}n^{-\frac{1}{2}}\right)$ (if at least one $\alpha_m \approx 1$ is small and thus $\alpha_{\min}$ is small) and almost as fast as $O\left(n^{-1}\right)$ (if α_m is large for all m and thus $\alpha_{\min}$ is large). For example, the latter is the case if all kernels have finite rank and also the convolution kernel is an example of this type.

Note that, of course, we could repeat the above discussion to obtain excess risk bounds for the case $p \geq 2$ as well, but since it is very questionable that this will lead to new insights, it is omitted for the sake of simplicity.

4.3.1 Discussion of Bounds

In this section, we discuss the rates obtained from the bound in Theorem 4.11 for the excess risk and compare them to the rates obtained using the global Rademacher complexity bound of Corollary 4.4. To somewhat simplify the discussion, we assume

that the eigenvalues satisfy $\lambda_j^{(m)} \leq dj^{-\alpha}$ (with $\alpha > 1$) for all m and concentrate on the rates obtained as a function of the parameters n, α, M, D and p, while considering other parameters fixed and hiding them in a big-O notation. Using this simplification, the bound of Theorem 4.11 reads

$$\forall t \in [p, 2] : \quad P(l_{\hat{f}} - l_{f^*}) = O\Big(\big(t^* D\big)^{\frac{2}{1+\alpha}} M^{1+\frac{2}{1+\alpha}\left(\frac{1}{t^*}-1\right)} n^{-\frac{\alpha}{1+\alpha}} \Big). \tag{4.22}$$

On the other hand, the global Rademacher complexity directly leads to a bound on the supremum of the centered empirical process indexed by $\mathcal{F}$ (see Appendix A) and thus also provides a bound on the excess risk (see, e.g., Bousquet et al., 2004). Therefore, using Corollary 4.4, wherein we upper bound the trace of each J_m by the constant B (and subsume it under the O-notation), we have a second bound on the excess risk of the form

$$\forall t \in [p, 2] : \quad P(l_{\hat{f}} - l_{f^*}) = O\Big(t^* D M^{\frac{1}{t^*}} n^{-\frac{1}{2}} \Big). \tag{4.23}$$

To start the discussion, we first consider the case where $p \geq (\log M)^*$, that is, the best choice in (4.22) and (4.23) is $t = p$. Clearly, if we hold all other parameters fixed and let n grow to infinity, the rate obtained through the local Rademacher analysis is better since $\alpha > 1$. However, it is also of interest to consider what happens when the number of kernels M and the radius D of the ℓ_p-ball can grow with n. In general, we have a bound on the excess risk given by the minimum of (4.22) and (4.23); a straightforward calculation shows that the local Rademacher analysis surpasses the global one whenever

$$\frac{M^{\frac{1}{p}}}{D} = O(\sqrt{n}).$$

Interestingly, we note that this "phase transition" does not depend on α (i.e. the "complexity" of the individual kernels), but only on p.

If $p \leq (\log M)^*$, the best choice in (4.22) and (4.23) is $t = (\log M)^*$. In this case taking the minimum of the two bounds reads

$$P(l_{\hat{f}} - l_{f^*}) \leq O\Big(\min\Big(D(\log M) n^{-\frac{1}{2}},\ \big(D \log M\big)^{\frac{2}{1+\alpha}} M^{\frac{\alpha-1}{1+\alpha}} n^{-\frac{\alpha}{1+\alpha}} \Big)\Big), \tag{4.24}$$

and the phase transition when the local Rademacher bound improves over the global one occurs for

$$\frac{M}{D \log M} = O(\sqrt{n}).$$

Finally, it is also interesting to observe the behavior of (4.22) and (4.23) as $\alpha \to \infty$. In this case, it means that only one eigenvalue is nonzero for each kernel, that is, each kernel space is one-dimensional. In other words, here we are in the case of "classical" aggregation of M basis functions, and the minimum of the two bounds reads

$$P(l_{\hat{f}} - l_{f^*}) \leq O\Big(\min\Big(Mn^{-1},\ \min_{t \in [p,2]} t^* D M^{\frac{1}{t^*}} n^{-\frac{1}{2}} \Big)\Big). \tag{4.25}$$

In this configuration, observe that the local Rademacher bound is $O(M/n)$ and does not depend on D, nor p, any longer; in fact, it is the same bound that one would obtain for the empirical risk minimization over the space of all linear combinations of the M base functions, without any restriction on the norm of the coefficients—the ℓ_p-norm constraint becomes void. The global Rademacher bound, on the other hand, still depends crucially on the ℓ_p-norm constraint. This situation is to be compared to the sharp analysis of the optimal convergence rates of convex aggregation of M functions obtained by Tsybakov (2003) in the framework of squared error loss regression, which are shown to be

$$O\left(\min\left(\frac{M}{n}, \sqrt{\frac{1}{n}\log\left(\frac{M}{\sqrt{n}}\right)}\right)\right).$$

This corresponds to the setting studied here with $D = 1$, $p = 1$ and $\alpha \to \infty$, and we see that the bound (4.24) recovers (up to log factors) the above bound and the related phase transition phenomenon.

4.3.2 Discussion of Assumption (U)

Assumption (U) is arguably quite a strong hypothesis for the validity of our results (needed for $1 \leq p \leq 2$), which was not required for the global Rademacher bound. A similar assumption was made in the recent work of Raskutti et al. (2010), where a related MKL algorithm using an ℓ_1-type penalty is studied, and bounds are derived that depend on the "sparsity pattern" of the Bayes function, i.e. how many coefficients w_m^* are non-zero. If the kernel spaces are one-dimensional, in which case ℓ_1-penalized MKL reduces qualitatively to standard lasso-type methods, this assumption can be seen as a strong form of the so-called Restricted Isometry Property (RIP), which is known to be necessary to grant the validity of bounds taking into account the sparsity pattern of the Bayes function.

In the present work, our analysis stays deliberately "agnostic" (or worst-case) with respect to the true sparsity pattern (in part because experimental evidence seems to point towards the fact that the Bayes function is not strongly sparse), correspondingly it could legitimately be hoped that the RIP condition, or Assumption (U), could be substantially relaxed. Considering again the special case of one-dimensional kernel spaces and the discussion about the qualitatively equivalent case $\alpha \to \infty$ in the previous section, it can be seen that Assumption (U) is indeed unnecessary for bound (4.25) to hold, and more specifically for the rate of M/n obtained through local Rademacher analysis in this case. However, as we discussed, what happens in this specific case is that the local Rademacher analysis becomes oblivious to the ℓ_p-norm constraint, and we are left with the standard parametric convergence rate in dimension M. In other words, with one-dimensional kernel spaces, the two constraints (on the $L^2(P)$-norm of the function and on the ℓ_p block-norm of the coefficients) appearing in the definition of local Rademacher complexity are essentially not simultaneously active. Unfortunately, it is clear that this property is not true anymore for kernels of higher complexity (i.e. with a non-trivial decay rate of the eigenvalues). This is a specificity of the kernel setting

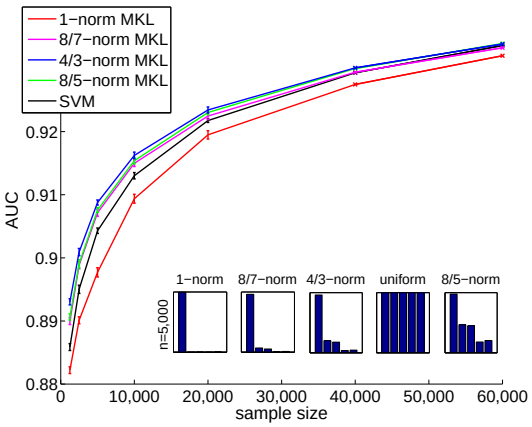
as compared to combinations of a dictionary of M simple functions, and Assumption (U) was in effect used to "align" the two constraints. To sum up, Assumption (U) is used here for a different purpose from that of the RIP in sparsity analyses of ℓ_1 regularization methods; it is not clear to us at this point if this assumption is necessary or if uncorrelated variables $\phi_(x)$ constitutes a "worst case" for our analysis. We did not succeed so far in relinquishing this assumption for $p \leq 2$, and this question remains open.

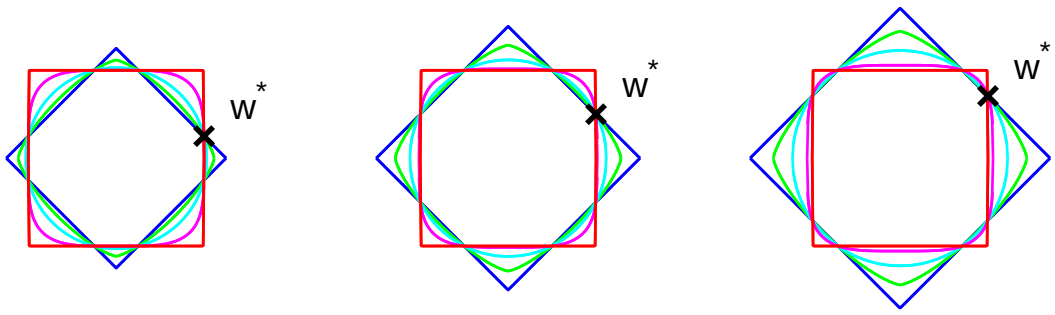
To our knowledge, there is no previous existing analysis of the ℓ_p-MKL setting for $p > 1$; the recent works of Raskutti et al. (2010) and Koltchinskii and Yuan (2010) focus on the case $p = 1$ and on the sparsity pattern of the Bayes function. A refined analysis of ℓ_p-regularized methods in the case of combination of M basis functions was laid out by Koltchinskii (2009), also taking into account the possible soft sparsity pattern of the Bayes function. Extending the ideas underlying the latter analysis into the kernel setting is likely to open interesting developments.

4.4 Why Can Learning Kernels Help Performance?

In this section, we give a practical application of the bounds of the previous section. We analyze the impact of the norm-parameter p on the accuracy of ℓ_p-norm MKL in various learning scenarios. We show that, depending on the underlying truth, any value of p can be optimal in practice. Since the trivial uniform-kernel-SVM baseline corresponds to $\ell_{p=2}$-norm MKL, this, in particular, shows that learning kernels can be beneficial. It also shows that whether or not this is the case depends on the *geometry* of the learning problem. This addresses the question "Can learning kernels help performance?" posed by Corinna Cortes in an invited talk at ICML 2009 (Cortes, 2009). Our answer is thus clearly affirmative: it is also in accordance with our empirical findings presented in the upcoming chapter.

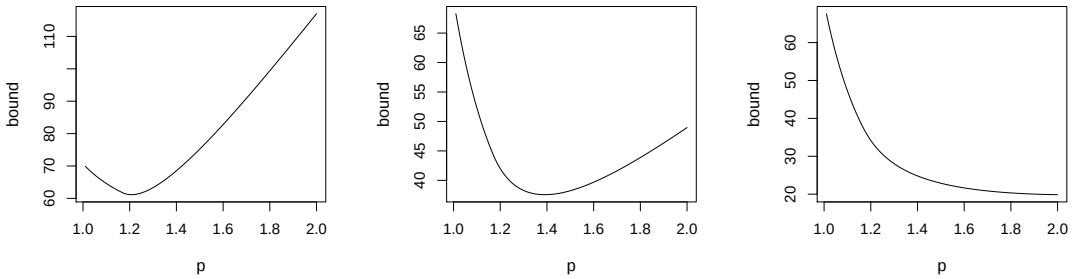
As will be shown in the application chapter of this thesis, there is empirical evidence that the performance of ℓ_p-norm MKL crucially depends on the choice of the norm parameter p. For example, the figure shown on the right-hand side, taken from Section 5.3 of this thesis, shows the result of a typical experiment with ℓ_p-norm MKL. We first observe that, as expected, ℓ_p-norm MKL enforces strong sparsity in the coefficients θ_m when $q = 1$, and no sparsity at all for $p = 2$, which corresponds to the SVM using a uniform kernel combination, while intermediate values of p enforce different degrees of soft sparsity (understood as the steepness of the decrease of the ordered coefficients θ_m). Crucially, the performance (as measured by the AUC criterion) is not





(a) $\beta = 2$ (b) $\beta = 1$ (c) $\beta = 0.5$

Figure 4.1: Two-dimensional illustration of the three analyzed learning scenarios, which differ in the soft sparsity of the Bayes hypothesis $\boldsymbol{w}^*$ (parameterized by β). LEFT: A soft sparse $\boldsymbol{w}^*$. CENTER: An intermediate non-sparse $\boldsymbol{w}^*$. RIGHT: An almost-uniformly non-sparse $\boldsymbol{w}^*$.



(a) $\beta = 2$ (b) $\beta = 1$ (c) $\beta = 0.5$

Figure 4.2: Results of the simulation for the three analyzed learning scenarios (which were illustrated in Figure 4.1). The value of the bound factor ν_t is plotted as a function of p. The minimum is attained depending on the true soft sparsity of the Bayes hypothesis $\boldsymbol{w}^*$ (parameterized by β).

monotonic as a function of q; $q = 1$ (sparse MKL) yields significantly worse performance than $q = \infty$ (regular SVM with sum kernel), but optimal performance is attained for some intermediate value of q.

The aim of this section is to relate the theoretical analysis presented here to the empirically observed phenomenon. This phenomenon can (at least partly) be explained on base of our excess risk bound obtained in the last section. To this end, we will analyze the dependancy of the excess risk bounds on the chosen norm parameter p. We will show that the optimal p depends on the geometrical properties of the learning problem and that in general—depending on the true geometry—any p can be optimal. Since our excess risk bound is only formulated for $p \leq 2$, we will limit the analysis to the range $p \in [1, 2]$.

To start with, note that the choice of p only affects the excess risk bound in the factor (cf. Theorem 4.11 and Equation (4.22))

$$\nu_p := \min_{t \in [p,2]} \left(D_p t^*\right)^{\frac{2}{1+\alpha}} M^{1+\frac{2}{1+\alpha}\left(\frac{1}{t^*}-1\right)}.$$

So we write the excess risk as $P(l_{\hat{f}} - l_{f^*}) = O(\nu_p)$ and hide all variables and constants in the O-notation for the whole section (in particular the sample size n is considered a constant for the purposes of the present discussion). It might surprise the reader that we consider the term depending on D although it seems from the bound that it does not depend on p. This stems from a inconspicuous fact that we have ignored in this analysis so far: D is related to the approximation properties of the class, i.e., its ability to attain the Bayes hypothesis. For a "fair" analysis we should take the approximation properties of the class into account.

To illustrate this, let us assume that the Bayes hypothesis belongs to the space $\mathcal{H}$ and can be represented by $\boldsymbol{w}^*$; assume further that the block components satisfy $\|\boldsymbol{w}_m^*\|_2 = m^{-\beta}$, $m = 1, \ldots, M$, where $\beta \geq 0$ is a parameter parameterizing the "soft sparsity" of the components. For example, the cases $\beta \in \{0.5, 1, 2\}$ are shown in Figure 4.1 for $M = 2$ and assuming that each kernel has rank 1 (thus being isomorphic to $\mathbb{R}$). If n is large, the best bias-complexity tradeoff for a fixed p will correspond to a vanishing bias, so that the best choice of D will be close to the minimal value so that $\boldsymbol{w}^* \in H_{p,D}$, that is, $D_p = ||\boldsymbol{w}^*||_p$. Plugging in this value for D_p, the bound factor ν_p becomes

$$\nu_p := \|\boldsymbol{w}^*\|_p^{\frac{2}{1+\alpha}} \min_{t \in [p,2]} t^{*\frac{2}{1+\alpha}} M^{1+\frac{2}{1+\alpha}\left(\frac{1}{t^*}-1\right)}.$$

We can now plot the value ν_p as a function of p for special choices of α, M, and β. We realize this simulation for $\alpha = 2$, $M = 1000$, and $\beta \in \{0.5, 1, 2\}$, which means we generate three learning scenarios with different levels of soft sparsity parameterized by β. The results are shown in Figure 4.2. Note that the soft sparsity of $\boldsymbol{w}^*$ is increased from the left hand to the right hand side. We observe that in the "soft sparsest" scenario ($\beta = 2$, shown on the left-hand side) the minimum is attained for a quite small $p = 1.2$, while for the intermediate case ($\beta = 1$, shown at the center) $p = 1.4$ is optimal, and finally in the uniformly non-sparse scenario ($\beta = 2$, shown on the right-hand side)

the choice of $p = 2$ is optimal (although even a higher p could be optimal, but our bound is only valid for $p \in [1, 2]$).

This means that if the true Bayes hypothesis has an intermediately dense representation, our bound gives the strongest generalization guarantees to ℓ_p-norm MKL using an intermediate choice of p. This is also intuitive: if the truth exhibits some soft sparsity but is not strongly sparse, we expect non-sparse MKL to perform better than strongly sparse MKL or the unweighted-sum kernel SVM.

4.5 Summary and Discussion

We justified the proposed ℓ_p-norm multiple kernel learning methodology from a theoretical point of view. Our analysis is built upon the established framework of local Rademacher complexities (Koltchinskii and Panchenko, 2002; Koltchinskii, 2006): we derived tight upper bounds on the local Rademacher complexity of the hypothesis class associated with ℓ_p-norm MKL, yielding excess risk bounds with fast convergence rates of the order $O(n^{-\frac{\alpha}{1+\alpha}})$, where α is the minimum eigenvalue decay rate of the individual kernels, thus being tighter than existing bounds for $\ell_{p>1}$-norm MKL, which only achieved slow convergence rates of the order $O(n^{-\frac{1}{2}})$. We also showed a lower bound on the local complexity, which, up to constants, matches the upper bounds, showing that our results are tight. For the upper bounds to hold, we used an assumption on the uncorrelatedness of the kernels; a similar assumption was also recently used by Raskutti et al. (2010), but in the different context of sparse recovery. We also remark that it would be interesting to even refine the analysis beyond local Rademacher complexities; Bartlett and Mendelson (2006) would be a good starting point for such an undertaking.

Beside reporting on the worst-case bounds, we also connected the minimal values of the bounds with the geometry of the underlying learning scenario (namely, the soft sparsity of the Bayes hypothesis), in particular, demonstrating that for a large range of learning scenarios ℓ_p-norm MKL attains a strictly "better" (i.e., lower) bound than classical ℓ_1-norm MKL and the SVM using a uniform kernel combination. This theoretically justifies using ℓ_p-norm MKL and multiple kernel learning in general. This is notable since even the tightest previous analyses such as the one carried out by Cortes et al. (2010a) were not able to answer the research question "Can learning kernels help performance?" (Cortes, 2009).

5 Empirical Analysis and Applications

In this chapter, we present an empirical analysis of ℓ_p-norm MKL. We first exemplify our experimental methodology on artificial data and real-world data, shedding light on the appropriateness of a high/low p in several learning scenarios that differ in the sparseness of the underlying Bayes hypothesis. We then apply the proposed ℓ_p-norm MKL on diverse, highly topical real-world problems from the domains of bioinformatics and computer vision that traditionally come with various heterogeneous feature groups / kernels and thus are very attractive for multiple kernel learning.

The main **contributions** in this chapter are the following:

- We investigate why learning kernels can help performance on artificial data, where we find the bounds well predicting the optimal value of the parameter p.
- We show that ℓ_p-norm MKL achieves accuracies in highly topical and central real-world applications from the domains of bioinformatics and computer vision that go beyond the state-of-the-art:
  (a) we apply ℓ_p-norm MKL to *protein fold prediction*, experimenting on the recently released *dingshen* data, yielding an accuracy improvement of 6%
  (b) we apply ℓ_p-norm MKL to *DNA transcription splice site (TSS) prediction*, training our large-scale implementation on 60,000 training examples and outperforming the uniform-sum kernel SVM *(which recently was confirmed to be leading in a comparison of 19 DNA TSS predictors)*
  (c) we apply ℓ_p-norm MKL to *visual image classification*, a computer vision application, significantly increasing the predictive accuracy for all concept classes and by 1.5 AP point in average.
- We introduce a methodology based on kernel alignments and the theoretical bounds to investigate/estimate our findings, i.e., the optimality of a particular p.

Parts of this chapter are based on:

M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Müller, and A. Zien. Efficient and accurate lp-norm multiple kernel learning. In *NIPS 2009*, pages 997–1005. MIT Press, 2009.

M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. Lp-norm multiple kernel learning. *Journal of Machine Learning Research (JMLR)*, 12:953–997, 2011.

A. Binder, S. Nakajima, M. Kloft, C. Müller, W. Wojcikiewicz, U. Brefeld, K.-R. Müller, and M. Kawanabe. Multiple kernel learning for object classification. Submitted to *IEEE TPAMI*. A preliminary version is published in *IBIS 2010*.

5.1 Goal and Experimental Methodology

Goal The goal of the application chapter of this thesis is to confirm the following hypotheses:

1. In practical applications that are characterized by multiple heterogeneous kernels / feature mappings, ℓ_p-norm MKL often yields more accurate prediction models

than ℓ_1-norm MKL, the SVM using a uniform kernel combination, and the single-kernel SVM with the kernel tuned by model selection.

2. For medium and large training set sizes n, the bounds presented in the previous chapter reflect the empirically best-performing parameter p (rendering them attractive for model selection) while for small n the Bayes hypothesis might be insufficiently approximated, leading to an overestimation of the optimal p.

5.1.1 Methodology

To investigate the validity of the above hypotheses, we experiment on cutting-edge data sets taken from diverse application domains. Thereby, we deploy the following experimental protocol:

1. Application description and goal First, the application's problem setting is described and the goal is defined; usually this is maximizing the prediction accuracy or an application-specific standard performance measure such as MCC, AUC, or AP (see below for definitions of this measures).

2. Experimental setup Second, we report on the employed data set and the experimental setup. Usually, we deploy random sampling of disjoint training, validation, and test sets, where the number of repetitions is chosen such that the standard errors indicate significance of the results (usually 100–250 repetitions). We always compare our results to the ones achieved by ℓ_1-norm MKL and the SVM using a uniform kernel combination. Often, the single-kernel-SVM performance is known from the literature, where it was either compared to ℓ_1-norm MKL or the SVM using an uniform kernel combination; in all of our experiments, the latter two baselines are contained in the set of models, thus allowing to also compare ℓ_p-norm MKL to the single-kernel SVM without actually recomputing it. However, if the single-kernel performance is not known from the literature, we compute this baseline. We compare various ℓ_p-norm MKL variants and deploy a local search for the optimal p. In all experiments, we optimize the relative duality gap up to a precision of 0.001.

3. Results The results of the experiment are shown in terms of an application-specific standard performance measure along with corresponding standard errors. In one case we use a benchmark data set that comes with a single test set; here, we compute the standard deviation by repeatedly and randomly splitting the test set in two parts of equal size.

4. Interpretation The kernel weights $\boldsymbol{\theta}$ as output by MKL are shown. This indicates which kernels contribute most to the final MKL solution. We also report on the alignments of the given kernel matrices with respect to the Frobenius scalar product. This can be of help in identifying redundant and complementary kernels.

5. BOUND We compute the factor of the theoretical bound that depends on the norm parameter p, thus being relevant for the estimation of the appropriateness of a small/large p. We focus on global Rademacher complexity as it does not involve the additional parameter α that is unknown in practice, which corresponds to taking $\alpha \approx 1$ in the local complexity (this can be seen as a conservative choice: the (empirical) global bound can directly be evaluated from the kernel matrices at hand without knowledge of the true spectra of the kernels). As discussed in Section 4.4, we require the Bayes hypothesis to be in the hypothesis set so that the relevant bound factor is $\nu_p^{\text{glob}} = \min_{t \geq p} \sqrt{t^*} \, \|\boldsymbol{w}_{\text{Bayes}}\|_{2,p} \, M^{1/t^*}$. Since this requires knowledge of the Bayes hypothesis $\boldsymbol{w}_{\text{Bayes}}$, we take the one hypothesis $\hat{\boldsymbol{w}}_{\text{MKL}}$ output by MKL that performed best in the experiments (using the empirically optimal p) as an approximation of $\boldsymbol{w}_{\text{Bayes}}$. Note that we expect this approximation getting tighter with increasing n. For the computation of $\hat{\boldsymbol{w}}_{\text{MKL}}$, we take medians over 100 runs; for the multi-class data sets the medians are also computed over the classes.

6. SUMMARY AND DISCUSSION Each experiment is concluded by a short discussion.

5.1.2 Evaluation measures

To assess the effectiveness of the compared methods, we use the following evaluation measures. Let $(x_t, y_t)_{t \in T}$ be a set of example/label test pairs and $f : \mathcal{X} \to \mathcal{Y}$ a hypothesis function; let $\mathcal{Y} = \{-1, 1\}$.

- The *test error* is defined as $\text{TE} = \sum_{t \in T} \mathbb{1}_{f(x_t) \neq y_t}$, where $\mathbb{1}$ denotes the indicator function.

- The *area under the receiver operating characteristic curve (AUC)* is defined as the integral of the receiver operating characteristic (ROC) curve. Thereby, the ROC curve is the true positive rate (fraction of correctly positively classified data points) as a function of the false positive rate (fraction of *in*correctly positively classified data points). For (multi-) kernel methods and linear methods the various values of the false positive rate correspond to translating the bias b. The AUC ranges in the interval $[0, 1]$, where 0.5 is attained by a completely random classifier (random label assignment); thus, reasonable classifiers should attain values higher than 0.5.

- The *Matthews correlation coefficient (MCC)* or *ϕ coefficient* is defined as

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \,,$$

  where FP and TP are the number of correctly and incorrectly positively classified data points, respectively, and FN and TN are defined analogously for negatively classified data points. The MCC ranges in the interval $[-1, 1]$ where $MCC = 0$ corresponds to a random prediction.

- *Average precision*: let $t_1, \ldots, t_n$ be the indexes corresponding to sorting the classifier outputs $f(x_t)$ in decreasing order. Denote the number of positive and negative examples by n_+ and n_-, respectively; define the precision at k by $p_k = \frac{1}{k} \sum_{i=1}^{k} \mathbb{1}_{y_{t_i}=1}$ (the number of correctly positively classified examples among the top-k ranked ones). Then, the average precision is defined as

$$\mathrm{AP} = \frac{1}{n_+} \sum_{k=1}^{n} p_k \mathbb{1}_{y_{t_k}=1}$$

  (thus it is the precision at k averaged over all k). The AP ranges in the interval $[0, 1]$ and somewhat depends on the skewness of the classes: a random classifier attains $\mathrm{AP} = \frac{n_+}{n_+ + n_-}$.

Probably the most standard measure in machine learning is the test error. However, in applications with unbalanced class sizes it is recommended to use one of the alternative measures, that is, MCC, AUC, or AP. Traditionally, some applications come with certain standard evaluation measures. For example, in image categorization, as an information retrieval application, AP is the standard measure and, in bioinformatics, MCC and AUC are frequently used.

Kernel aligments As a tool for the explorative analysis of the obtained results, we also employ the *kernel alignment* introduced by Cristianini et al. (2002), which measures the similarity of two matrices and can be interpreted as a (hyper-) kernel acting on the space of kernel functions (Ong et al., 2005). Let K and $\tilde{K}$ be two kernel matrices corresponding to kernels k and $\tilde{k}$, respectively; then, the alignment of K and $\tilde{K}$ is defined as the cosine of the angle between K and $\tilde{K}$:

$$\mathcal{A}(K, \tilde{K}) := \frac{\langle K, \tilde{K} \rangle_F}{\|K\|_F \|\tilde{K}\|_F}, \tag{5.1}$$

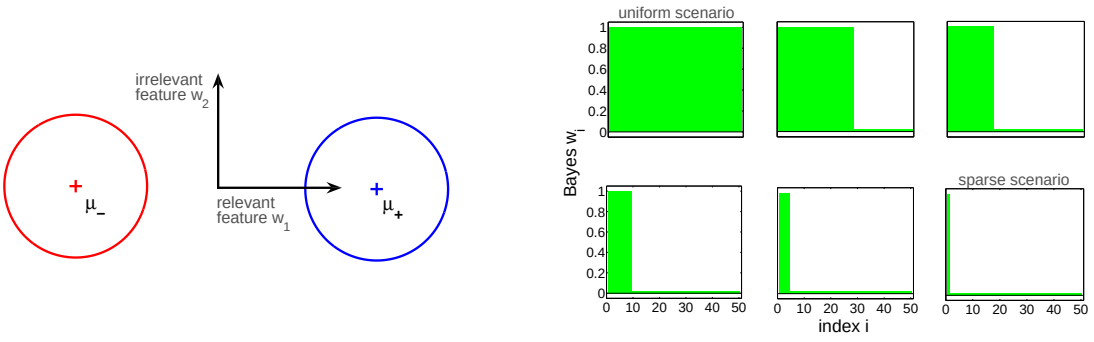
where $\langle K, \tilde{K} \rangle_F := \sum_{i,j=1}^{n} k(x_i, x_j) \tilde{k}(x_i, x_j)$ denotes the Frobenius scalar product and $\|K\|_F := \sqrt{\langle K, K \rangle_F}$ the corresponding norm.

In many applications, centering of the kernels is recommended before computing the alignment (Cortes et al., 2010b); for example, SVMs and many other kernel-based learning algorithms are invariant against mean shifts in the corresponding Hilbert spaces. Centering K in the corresponding feature space can easily be achieved by computing the product HKH, where $H := I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ with I being the identity matrix of size n and $\mathbf{1}$ a column vector of all ones.

5.2 Case Study 1: Toy Experiment

We now present a toy experiment that serves the following purposes:

- empirically confirming some of our claims in a controlled environment, that is, one where we know the underlying distribution generating the data



(a) Illustration (b) Scenarios

Figure 5.1: The toy experiment: experimental design.

- exemplary application of the experimental protocol described in the previous section.

The claims we would like to confirm are the following:

1. The bounds reflect the empirically optimal p if the Bayes hypothesis is known.
2. The pairwise kernel alignments can be successfully used to explore dependancies in the kernel set.
3. The choice of the norm parameter p can be crucial for the generalization performance of ℓ_p-norm MKL.
4. The optimality of a particular p depends on the underlying geometry (i.e, the sparsity) of the underlying Bayes hypothesis.

Experimental setup We construct six artificial data sets in which we vary the degree of sparsity in the true Bayes hypothesis $\boldsymbol{w}$. For each data set, we generate an n-element, balanced sample $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ from two $d = 50$-dimensional isotropic Gaussian distributions with equal covariance matrices $C = I_{d\times d}$ and equal, but opposite, means $\mu_+ = \frac{\rho}{\|\boldsymbol{w}\|_2}\boldsymbol{w}$ and $\mu_- = -\mu_+$. Thereby, $\boldsymbol{w}$ is a binary vector, i.e., $\forall i : w_i \in \{0, 1\}$, encoding the true underlying data sparsity as follows. Zero components $w_i = 0$ clearly imply identical means of the two classes' distributions in the ith feature set; hence, the latter does not carry any discriminating information. In summary, the fraction of zero components, $\mathit{sparsity}(\boldsymbol{w}) = 1 - \frac{1}{d}\sum_{i=1}^d w_i$, is a measure for the feature sparsity of the learning problem. This is illustrated in Figure 5.1 (a).

We generate six different $\boldsymbol{w}$ that differ in their value of $\mathit{sparsity}(\boldsymbol{w})$; see Figure 5.1 (b), which shows bar plots of the $\boldsymbol{w}$ of the various scenarios considered. For each of the $\boldsymbol{w}$ we generate $m = 250$ data sets $\mathcal{D}_1, \ldots, \mathcal{D}_m$ fixing $\rho = 1.75$. Then, each feature is input into a linear kernel and the resulting kernel matrices are multiplicatively normalized as described in Section 3.3.1. Hence, $\mathit{sparsity}(\boldsymbol{w})$ gives the fraction

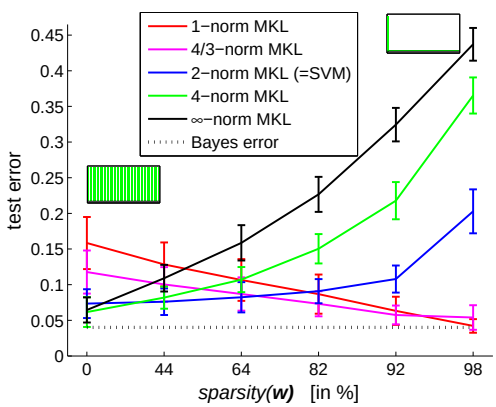
of noise kernels in the working kernel set. Next, classification models are computed by training ℓ_p-norm MKL for $p = 1, 4/3, 2, 4, \infty$ on each $\mathcal{D}_i$. Soft margin parameters C are tuned on independent $1,000$-elemental validation sets by grid search over $C \in \left\{10^i \,\middle|\, i = -4, -3.5, \ldots, 0\right\}$ (optimal Cs are attained in the interior of the grid). The relative duality gaps were optimized up to a precision of 10^{-3}. The simulation is realized for $n = 50$. We report on test errors evaluated on $1,000$-elemental independent test sets.

Results The results in terms of test errors are shown in Figure 5.2 (a). As expected, ℓ_1-norm MKL performs best and reaches the Bayes error in the sparsest scenario, where only a single kernel carries the whole discriminative information of the learning problem. However, in the other scenarios it mostly performs worse than the other MKL variants. This is remarkable because the underlying ground truth, i.e., the vector $\boldsymbol{w}$, is sparse in all but the uniform scenario. In other words, selecting this data set may imply a bias towards ℓ_1-norm. In contrast, the vanilla SVM using a uniform kernel combination performs best when all kernels are equally informative; however, its performance does not approach the Bayes error. In contrast, the truly uniform ℓ_∞-norm MKL variant succeeds in approaching the Bayes error although it does not reach it completely. However, it should be kept in mind that such a uniform scenario might be quite artificial. The non-sparse $\ell_{4/3}$-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 $\ell_{4/3}$-norm MKL is the most robust MKL variant, achieving test errors of less than 12% in all scenarios. Tuning the sparsity parameter p for each experiment, ℓ_p-norm MKL achieves low test error across all scenarios.

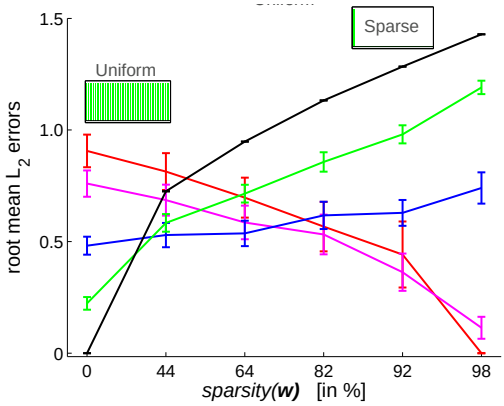
Interpretation We also consider the weights $\hat{\boldsymbol{w}}_{\mathrm{MKL}}$ output by MKL and compare them to the true underlying $\boldsymbol{w}_{\mathrm{Bayes}}$ by computing the root mean ℓ_2 (model) errors, $\mathrm{ME}(\hat{\boldsymbol{w}}_{\mathrm{MKL}}) := \|\zeta(\hat{\boldsymbol{w}}_{\mathrm{MKL}}) - \zeta(\boldsymbol{w}_{\mathrm{Bayes}})\|_2$, where $\zeta(\boldsymbol{v}) = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|_2}$. The results are shown in Figure 5.2 (b). We observe that the model errors reflect the corresponding test errors well. This observation can be explained by model selection favoring strongly regularized hypotheses for such a small n, leading to the observed agreement between test error and model error.

We are also interested in whether pairwise kernel alignments can be used as a measure for the dependancy of the kernels. For the toy experiment, we know the true dependancies: each noise kernel is independent of any other kernel by construction (regardless of whether the latter is a noise kernel as well or an information-carrying one); on the other hand, we would expect the informative kernels being mutually aligned since they are correlated via the labels (but only slightly so because the conditional distributions of the features are independent; this is a specialty of our data generation setup). The empirical kernel alignments are shown in Figure 5.2 (c). Indeed, as expected, we can see from the plot that overall the alignments are rather small although we can observe moderate alignments between informative kernels. This indicates the usefulness of this explorative method in practice.

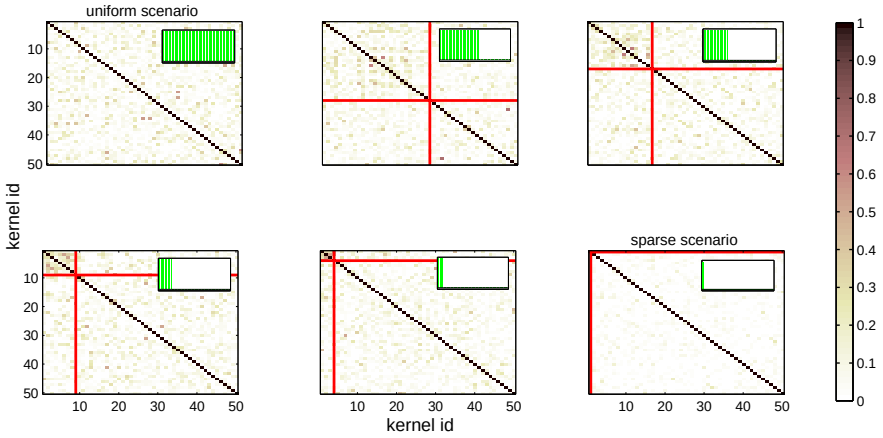
Bound We evaluate the theoretical bound factor for the various scenarios, exploiting the fact that the Bayes hypothesis is completely known to us. To analyze whether the p that are minimizing the bound are reflected in the empirical result, we compute the test



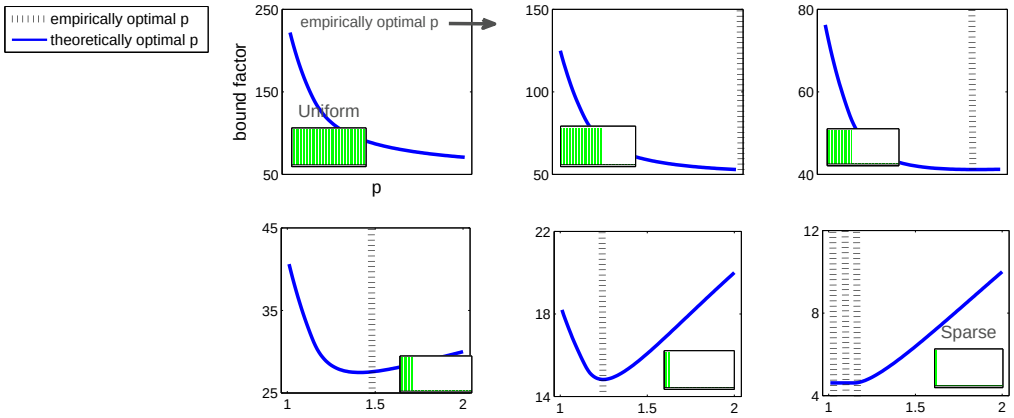
(a) Test errors



(b) Model errors



(c) Alignments



(d) Bound

Figure 5.2: The toy experiment: results and analysis.

errors of the various MKL variants again, using the setup above except that we employ a much finer grid for finding the optimal p. The results are shown in Figure 5.2 (d). We can observe that the minima of the bounds clearly reflect the p that are found to also work well empirically: In the sparsest scenario (shown on the lower right-hand side), the bound predicts $p \in [1, 1.14]$ to be optimal and indeed, in the experiments, all $p \in [1, 1.15]$ performed best (and equally well) while the next higher p contained in our grid, namely $p = 1.19$, already has a slightly (but significantly) worse test error—in striking match with our bounds. In the second sparsest scenario, the bound predicts $p = 1.25$ and we empirically found $p = 1.26$. In the non-sparse scenarios, intermediate values of $p \in [1, 2]$ are optimal (see Figure for details)—again we can observe a good accordance of the empirical and theoretical values. In the extreme case, i.e., the uniform scenario, the bound indicates a p that lies well beyond the valid interval of the bound (i.e., $p > 2$) and this is also what we observed empirically: $p \in [4, \infty]$ worked best in our experiments.

Summary and discussion We can conclude that the results confirm our two claims stated at the beginning of this subsection: first, the theoretical bounds reflect the empirically observed optimal p in the idealized setup where we know the Bayes hypothesis; second, we showed that kernel alignments reflect the dependancy structure of the kernel set; third, the choice of the norm parameter p can be crucial to obtaining high test set accuracy (for example, in the sparsest scenario ℓ_∞-norm MKL has 43% test error while ℓ_1-norm MKL reaches the Bayes error with an test error of 4%; cf. Figure 5.2(a)).

Last, we observed that the optimality of a particular p strongly depends on the geometry of the learning task: the sparsity of the underlying Bayes hypothesis $\boldsymbol{w}$. If $\boldsymbol{w}$ contains many irrelevant features, a small p is beneficial whereas intermediate or large p otherwise. This raises the question into which scenario practical applications fall. For example, do we rather encounter a "sparse" or non-sparse scenario in bioinformatics? This will be investigated in the upcoming real-world experiments. Our answer is somewhat mixed: we will show that, for example, bioinformatics is a domain too diverse to be categorized into a single learning scenario (e.g., sparse). In fact, we will present diverse bioinformatics applications covering the whole range of scenarios: sparse, intermediate non-sparse, and uniform (see Section 5.4). However, as observed from the toy experiment, by appropriately tuning the norm parameter, ℓ_p-norm MKL can prove robust in all scenarios.

5.3 Case Study 2: Real-World Experiment (TSS)

In this section, we study transcription splice site (TSS) detection, a highly topical, large-scale bioinformatics application. The purposes of this section are the following:

1. studying the prediction accuracy of ℓ_p-norm MKL for TSS detection
2. exemplarily carrying out the proposed methodology for a real-world application
3. investigating the impact of the sample size n on the performance of ℓ_p-norm MKL

(as we have access to 93,500 data points for this application)

4. investigating the impact of the sample size on the estimation of the optimal p by means of the theoretical bounds.

Application description and goal This experiment aims at detecting transcription start sites (TSS) of RNA Polymerase II binding genes in genomic DNA sequences. The accurate detection of the transcription start site (see the figure to the right for an illustration of the start site) is crucial to identifying genes and their promoter regions and can be regarded as a first step in deciphering the key regulatory elements that determine transcription. Those transcription start sites are located in the core promoter region and, for the majority of species, their localization must be achieved without the help of massive sequencing: for some species, including human, large scale sequencing projects of complete mRNAs have been undertaken, but many low copy genes still evade being sequenced, leaving a huge demand for accurate ab initio TSS prediction algorithms. Consequently, a fairly large number of TSS finders, exploiting 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), have been developed, which called for a comparison. To this end, the recent study of Abeel et al. (2009) compared 19 state-of-the-art promoter prediction programs in terms of their predictive accuracy. Here, the winning program of Sonnenburg et al. (2006b), entitled ARTS, was perceived to be leading.

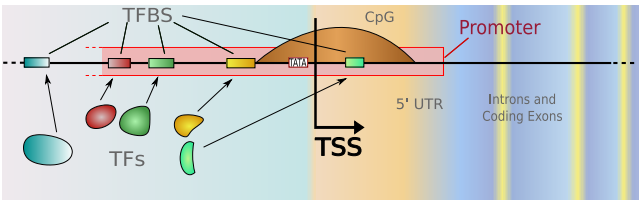


Figure 5.3: Figure taken from Alberts et al. (2002).

Interestingly, ARTS deploys an SVM using a uniform combination of five heterogeneous kernels capturing various aspect of the promoter region, rendering ARTS attractive for the application of ℓ_p-norm MKL. In this thesis, we study whether ℓ_p-norm MKL can be used to even further improve this cutting-edge accuracy. Note that a comparison to single-kernel SVMs was already carried out in Sonnenburg et al., 2006b (see Tables 2 and 3), showing that the uniform kernel combination is more accurate than the best single-kernel SVM, rendering another comparison unnecessary.

Experimental setup 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 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 3.3.1. We reserve

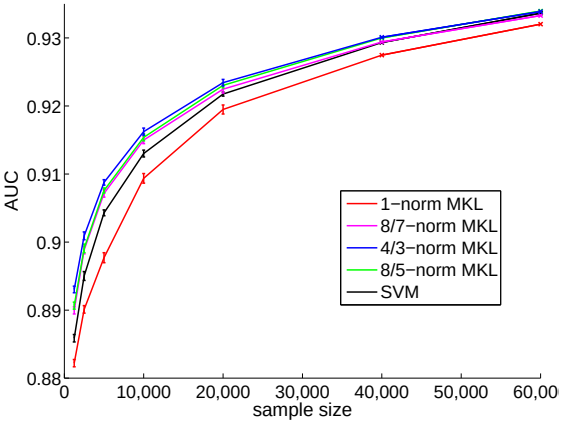
13,000 and 20,000 randomly drawn instances for validation and test sets, respectively, and use the remaining 60,000 as the training pool. Soft margin parameters C are tuned on the validation set by grid search over $C \in 2^{[-2,-1,\ldots,5]}$ (optimal Cs are attained in the interior of the grid). Figure 5.4 (a) 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.

Results Regardless of the sample size, ℓ_1-norm MKL is significantly outperformed by the SVM using a uniform kernel combination. In contrast, non-sparse MKL achieves significantly higher AUC values than the SVM using a uniform combination of kernels for sample sizes up to 40,000. The scenario is well suited for $\ell_{4/3}$-norm MKL which performs best. Finally, for 60,000 training instances, regularization becomes less and less important so that all methods except ℓ_1-norm MKL perform similarly with, again, $\ell_{4/3}$-norm MKL performing best among all prediction models. This superior performance of non-sparse MKL regardless of the sample size is remarkable and of significance for the application domain: as indicated above, the method using the unweighted sum of kernels has recently been confirmed to be leading in a comparison of 19 state-of-the-art promoter prediction programs and our experiments suggest that this accuracy can be further increased by non-sparse MKL, especially when data is spare. But obtaining data for this application is costly: even the most modern sequencing techniques used in the domain are not completely automatic and need human input by an biology expert. Therefore, our method is especially appealing for genomes that are hardly sequenced.

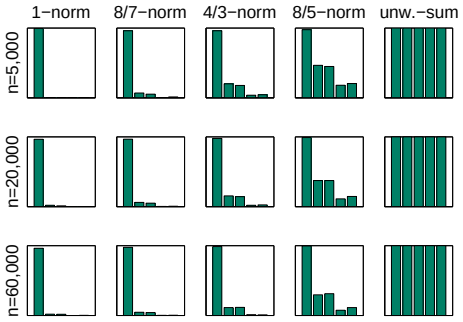
Interpretation To further explore possible reasons for optimality of a non-sparse ℓ_p-norm in the above experiments, we also recall the single-kernel performance of the kernels: TSS signal 0.89, promoter 0.86, 1st exon 0.84, angles 0.55, and energies 0.74, for fixed sample size $n = 2000$. This indicates that there are three highly and two moderately informative kernels. While non-sparse MKL distributes the weights over all kernels (see Figure 5.4 (b)), 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 compute the pairwise alignments[8] of the centered kernel matrices, i.e., $\mathcal{A}(i,j) = \frac{<K_i,K_j>_F}{\|K_i\|_F\|K_j\|_F}$, with respect to the Frobenius dot product (e.g., Golub and van Loan, 1996). The computed alignments are shown in Figure 5.4 (c). One can observe that the three relevant kernels are highly aligned, as was to be expected, since they are correlated via the labels. However, the energy kernel shows only slight correlations with the remaining kernels, which is surprisingly little compared to its single kernel performance (AUC=0.74). We conclude that this kernel carries complementary 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. 5.4 (b), and the reason for the empirical success of non-sparse MKL on this data set.

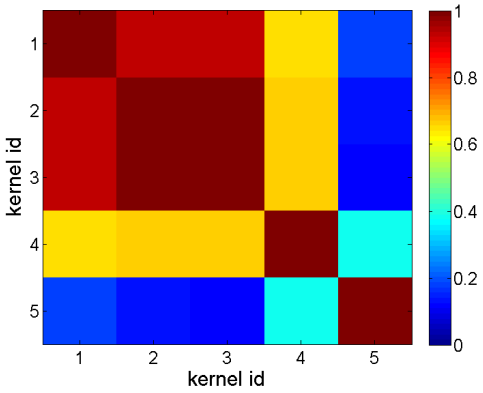
[8]The alignments can be interpreted as empirical estimates of the Pearson correlation of the kernels (Cristianini et al., 2002).



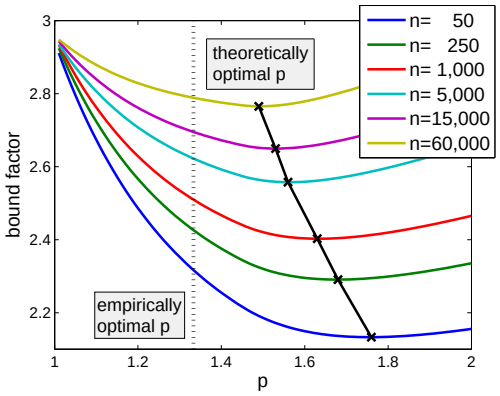
(a) Results (in AUC)



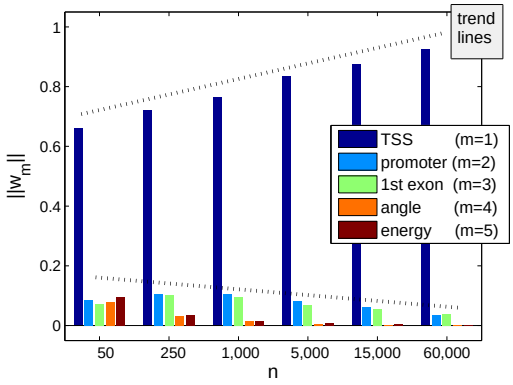
(b) MKL weights $\boldsymbol{\theta}$



(c) Alignments



(d) Bound



(e) MKL block weights $\|\boldsymbol{w}_m\|$

Figure 5.4: The transcription splice site (TSS) detection experiment: results and analysis.

Bounds For various training set sizes n, we compute the bound factor as a function of p. The results are shown in Figure 5.4 (d). We can observe the optimal p, as predicted by the theoretical bounds, apparently converging towards the empirically optimal p (which is $p = 4/3$) when increasing the training set size n. This results from the approximation of the Bayes hypothesis being tighter for large n: we compute $\hat{\boldsymbol{w}}_{\text{MKL}}$ as output by MKL for various n (shown in Figure 5.4 (e)) and observe that, with increasing n, MKL tends to discard the angle and energy kernels; the latter two kernels are of finite rank. Their limited complexity renders them less effective for large n. We thus conjecture that the Bayes hypothesis is likely to focus on the first three kernels (with an emphasis on the TSS kernel, which also empirically performs best). In this sense, it is not surprising that the bound "overestimates" the optimal p: since $\hat{\boldsymbol{w}}_{\text{MKL}}$ is "too uniform" compared to the Bayes-$\boldsymbol{w}$, the bound (which has $\hat{\boldsymbol{w}}_{\text{MKL}}$ as an input) leads to overly high estimates of the optimal p. However, it is interesting that already for $n = 60,000$ the bound gives a good estimate ($p = 1.48$) of the empirically optimal $p = 1.33$.

Summary and discussion Exemplarily, we carried out the proposed experimental methodology for transcription splice site detection, a cutting-edge bioinformatics application. The data set we experimented on was also used in the previous study of Sonnenburg et al. (2006a) and comprises 93,500 instances. We showed that regardless of the sample size, ℓ_p-norm MKL can improve the prediction accuracy over ℓ_1-norm MKL and the SVM using a uniform kernel combination (which itself outperforms the single-kernel SVM using the kernel determined by model selection). This is remarkable and of significance for the application domain: the method using a uniform kernel combination was recently confirmed to be the most accurate TSS detector in a comparison of 19 state-of-the-art models Abeel et al. (2009). Furthermore, we observed that the choice of n had no impact on the empirical optimality of a particular p. We also evaluated the theoretical bounds and found that with increasing n they give a more and more accurate estimation of the optimal p through an increasingly tight approximation of the Bayes hypothesis.

5.4 Bioinformatics Experiments

In this section, we apply ℓ_p-norm MKL to diverse, highly topical applications taken from the domain of bioinformatics. Many bioinformatics applications are too complex to be described by a single type of feature descriptors; for example, finding splice sites in genes can only be done moderately well on the raw sequence information—incorporating complementary information such as binding energies or twistedness of DNA increases the chance of finding a splice site. A challenge is here to deal with the additional noise introduced by those so-called *weak features*, making it very appealing for multiple kernel learning.

We recall the toy experiment, which showed that the amount by which ℓ_p-norm MKL is beneficial in such applications crucially depends on the sparsity/uniformness of the learning task at hand. We are therefore interested in finding out into which

scenario particular bioinformatics applications fall. We show that bioinformatics is a domain so diverse that we can encounter scenarios ranging from sparse over non-sparse to uniform ones. Nevertheless, appropriately tuning the norm parameter p, ℓ_p-norm MKL proves robust in all applications considered.

5.4.1 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, classical, ℓ_1-norm MKL could raise the predictive accuracy significantly above that of vanilla SVMs using a uniform kernel combination or the best kernel determined by model selection—thereby also surpassing 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 thesis, we investigate the performance of ℓ_p-norm MKL on the same 4 data sets.

Application description and goal There are several possible locations for a *protein* to reside in a cell: lysosome, mitochondrion, nucleus, etc. (see the Figure to the right for a list of possible locations). Predicting such a location is the goal of *protein subcellular localization*. The accurate localization of proteins in cells is important because the location of a protein is very closely connected to its function. This is especially relevant in pharmacology because the knowledge of the function of proteins is crucial when designing new drugs. Unfortunately, the manual localization of the protein is a time-consuming task. This renders bioinformatics approaches appealing.

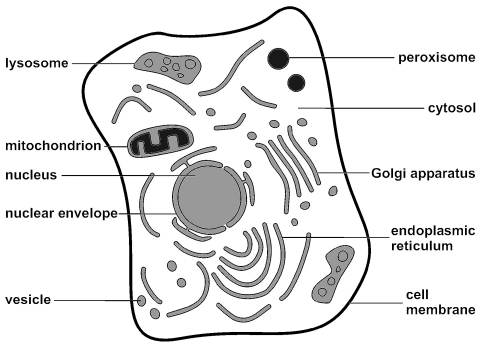


Figure 5.5: Figure taken from Schölkopf et al. (2004).

Experimental setup We downloaded the kernel matrices of all 4 data sets from `http://www.fml.tuebingen.mpg.de/raetsch/suppl/protsubloc/`. The kernel matrices are multiplicatively normalized as described in Section 3.3.1. 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 $p \in \{1, 1.01, 1.03, 1.07, 1.14, 1.33, 1.6, 2\}$ and regularization constants $C \in \{1/32, 1/8, 1/2, 1, 2, 4, 8, 32, 128\}$. For each parameter setting (p, C), we train ℓ_p-norm MKL using a 1-vs-rest strategy on the training set. The predictions on the test set are then evaluated w.r.t. average (over the classes) MCC (Matthews correlation coefficient). As we are only interested in the influence of the norm on the performance,

we forbear proper cross-validation (the so-obtained systematical error affects all norms equally). Instead, for each of the 30 data splits and for each p, the value of C that yields the highest MCC is selected. Thus we obtain an optimized C and MCC value for each combination of data set, split, and norm p. For each norm, the final MCC value is obtained by averaging over the data sets and splits (i.e., C is selected to be optimal for each data set and split).

Note that the results for the best single-kernel SVMs were reported in Ong and Zien, 2008 (see Figure 3): the single-kernel SVM was shown to be outperformed by both the SVM using a uniform kernel combination and ℓ_1-norm MKL on all four data sets.

Results The results, shown in Figure 5.6 (a), indicate that, indeed, with proper choice of a non-sparse regularizer, the accuracy of ℓ_1-norm can be recovered. On the other hand, non-sparse MKL can approximate ℓ_1-norm MKL arbitrarily closely, and thereby approach the same results. However, even when 1-norm is clearly superior to ∞-norm, as demonstrated for these 4 data sets, it is possible that intermediate norms perform even better. As the figure shows, this is indeed the case for the PSORT data sets, albeit only slightly and not significantly so.

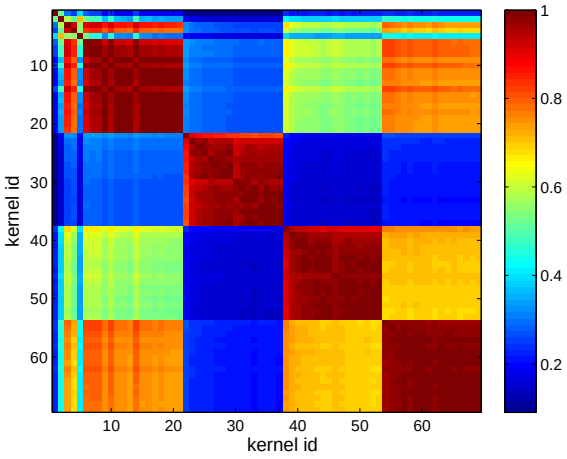
Interpretation At this point, we can remark that the superior performance of $\ell_{p\approx 1}$-norm MKL in this setup is not surprising. As the kernel alignment plots indicate (see Figure 5.6 (b)), there are four sets of 16 kernels each, in which each kernel picks up very similar information: they only differ in number and placing of gaps in all substrings of length 5 of a given part of the protein sequence. The situation is roughly analogous to considering (inhomogeneous) polynomial kernels of different degrees on the same data vectors. This means that they carry large parts of overlapping information. By construction, also some kernels (those with less gaps) also have access to more information (similar to higher degree polynomials including low degree polynomials). Furthermore, Ong and Zien (2008) studied single kernel SVMs for each kernel individually and found that in most cases the 16 kernels from the same subset perform very similarly. This means that each set of 16 kernels is highly redundant and the excluded parts of information are not very discriminative. This renders a non-sparse kernel mixture ineffective. We conclude that ℓ_1-norm must be the best prediction model.

Bound Figure 5.6 (c) shows the results of the bound simulation. We observe from the figure that the theoretical bounds predict $p = 1.2$, which is close to the empirically measured optimal $p = 1$, although it does not exactly match that value due to the approximation of the Bayes hypothesis involved (see discussion in the previous case study (TSS)). However, we can see from the figure that the curvature of the bound increases with n so that large values of p are more strongly excluded by the bound when n is larger. This might indicate a trend towards a sparser theoretically optimal p when increasing n as also observed in the TSS case study—however, to further investigate this hypothesis, we would need access to additional data, which, unfortunately, is not available for the application at hand.

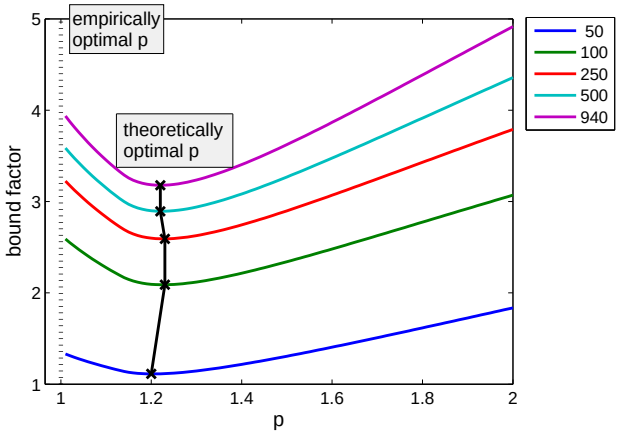
Summary and discussion We applied the proposed ℓ_p-norm MKL to the subcellular localization of proteins, which is one of most successful applications of computer systems

ℓ_p-norm	1	1.01	1.03	1.07	1.14	1.33	1.6	1.8	SVM
plant	**8.18**	8.22	8.20	8.21	8.43	9.47	11.00	11.61	11.85
std. err.	±0.47	±0.45	±0.43	±0.42	±0.42	±0.43	±0.47	±0.49	±0.60
nonpl	**8.97**	9.01	9.08	9.19	9.24	9.43	9.77	10.05	10.33
std. err.	±0.26	±0.25	±0.26	±0.27	±0.29	±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	13.33
std. err.	±0.35	±0.34	±0.34	±0.34	±0.33	±0.32	±0.32	±0.34	±0.35
psortPos	13.07	**13.01**	13.41	13.17	13.25	14.68	15.55	16.43	17.63
std. err.	±0.66	±0.63	±0.67	±0.62	±0.61	±0.67	±0.72	±0.81	±0.80

(a) Results (in MCC)



(b) Alignments



(c) Bound

Figure 5.6: The protein subcellular localization experiment: results and analysis.

to biology. We confirmed the results of Ong and Zien (2008) in that MKL greatly helps performance in this application, which we found to fall into the category of a sparse learning scenario: ℓ_1-norm MKL lead to an improvement of up to 4.5% points MCC over the SVM using a uniform kernel combination. The optimality of a low p is also reflected in the bounds, which predicted $p = 1.2$.

5.4.2 Protein Fold Prediction—A Non-Sparse Scenario

In this section, we experiment on the *dingshen* data which was used in the previous study of Campbell and Ying (2011).

Application description and goal Proteins are the functional molecular components inside cells. The so-called messenger RNA is transcribed from a genetic sequence that codes for a particular protein. An important step in this process is folding, in which the protein forms its final three-dimensional structure. Understanding the three-dimensional structure of a protein can give insight into its function. For example, if the protein is a drug target, knowledge of its structure is important in the design of small molecular inhibitors which would bind to, and disable, the protein. Advances in gene sequencing technologies have resulted in a large increase in the number of identified sequences that code for proteins (Campbell and Ying, 2011, p. 63). However, there has been a much slower increase in the number of known three-dimensional protein structures. This motivates computer-aided prediction of the structure of a protein from sequence and other data. In the present study, we consider the sub-problem of structure prediction, in which the predicted label is over a set of fold classes. The fold classes are a set of structural components, common across proteins, which give rise to the overall three-dimensional structure.

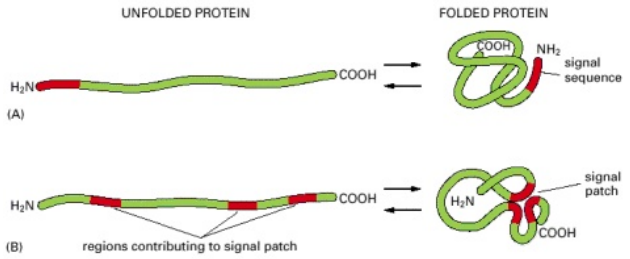


Figure 5.7: Figure taken from Alberts et al. (2002).

Experimental setup We obtained the dingshen data set, including a fixed training and test split, from Colin Campbell. The dingshen data set consists of 27 fold classes with 313 proteins used for training and 385 for testing. There are a number of observational features relevant to predicting the fold class; in this study, we used 12 different informative data types. These included the RNA sequence and various physical measurements such as hydrophobicity, polarity, and van-der-Waals volume, resulting in 12 kernels. We precisely replicate the experimental setup of Campbell and Ying (2011): we use the train/test split provided by Colin Campbell and carry out MKL via one-vs.-rest SVMs to deal with the multiple classes; we report on test set accuracy. We perform model selection by cross validation on the training set over $C \in 10^{[-4,-3.5,\ldots,4]}$. Since we only have a single test set at hand, we compute the standard deviation by 20 times

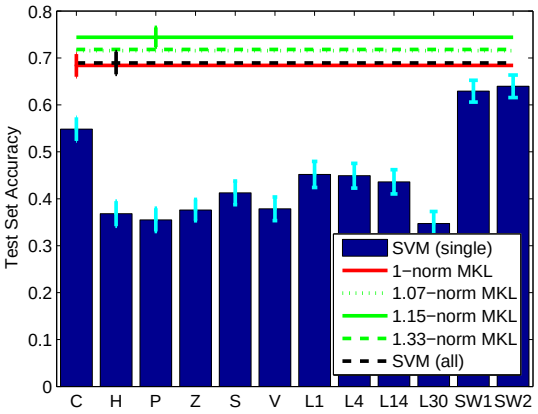
randomly splitting the test set in two parts of equal size and computing the standard deviations (shown as slim vertical bars).

Results The results are shown in Figure 5.8 (a). Flat vertical bars show the test set accuracy of the single-kernel SVMs: for example, H is Hydrophobicity, P is Polarity, V is van der Waals volume. Slim horizontal bars show the performances of ℓ_p-norm and the SVM using a uniform kernel combination. We observe that the best single-kernel SVM is the one using the SW2-kernel, having a test set accuracy of 64.0%; in contrast, the SVM using a uniform kernel combination achieves the substantially higher accuracy of 68.9%, which is slightly better than the 68.4% reached by ℓ_1-norm MKL. Interestingly, there is a huge improvement in using non-sparse $\ell_{p>1}$-norm MKL: the best performing norm is $p = 1.14$, which has an impressive accuracy of 74.4% and significantly outperforms the SVM baselines.

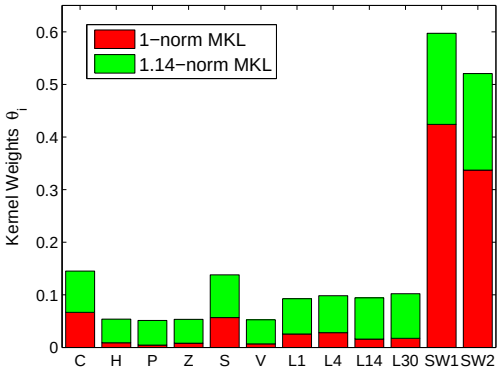
Interpretation Figure 5.8 (b) gives the values of the kernel coefficients $\boldsymbol{\theta}$. We observe that ℓ_1-norm MKL emphasizes on the SW1 and SW2 kernels, which also have the highest single-kernel performance. The $\ell_{p>1}$-norm variants yield precisely the same "ranking" of weights θ_i but distribute the weights among the kernels more strongly. Generally, the kernel combinations output by MKL nicely reflect the true performances as determined by the single-kernel SVMs. The superior performance of winning $\ell_{1.14}$-norm MKL compared to ℓ_1-norm MKL and the SVM using a uniform kernel combination indicates that although all 12 types of data are relevant, they are not equally so. For example, the features SW1 and SW2, which are based on sequence alignments, appear to be more informative than the others.

To further analyze the result, we compute the pairwise kernel alignments shown in Figure 5.8 (c). One can see from the figure that the Kernels L1–L30 and SW1–SW2 correlate quite strongly. This resembles the similarity in the generation process of those kernels (they differ by different parameter values). However, the other kernels correlate surprisingly little—this indicates that here complementary information is contained in the kernels. Therefore, discarding or overly downgrading one of those kernels can be disadvantageous, which explains the poor ℓ_1-norm MKL performance. On the other hand, we know from the single-kernel performances that not all kernels are equally informative, which explains the rather bad performance of the uniform-combination SVM. We conclude that an intermediate norm must be optimal—and this is also what we observe in terms of test errors.

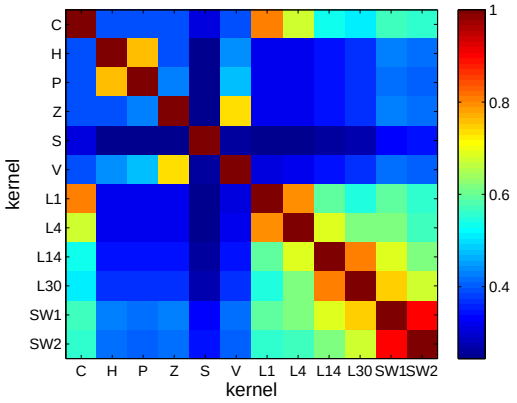
Bound The bound factor, shown as a function of p in Figure 5.8 (d) indicates a theoretically optimal p too high compared to the empirically optimal one. This may stem from the small sample that is employed here. Indeed, in the application at hand, we face 313 training examples from 27 classes, that is, only 11.6 examples per class in average. As shown in Case Study 2, small samples can result in an underestimation of the true underlying sparsity of the task and this is probably also what happens here. This hypothesis is further supported by the fact that we can observe the curvature of the bound increasing with n. This might indicate that if we had access to a larger sample size (resulting in a tighter approximation of the Bayes hypothesis), a lower p might be optimal in the bound. However, for the limited data at hand, we cannot fully



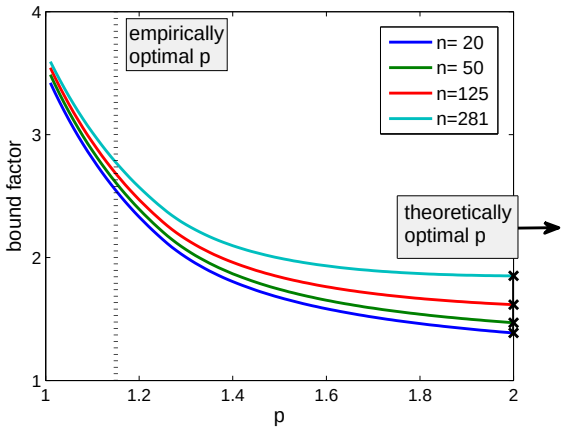
(a) Results (accuracy)



(b) MKL weights $\boldsymbol{\theta}$



(c) Alignments



(d) Bound

Figure 5.8: The protein fold prediction experiment: results and analysis.

verify this hypothesis.

Summary and discussion We studied ℓ_p-norm MKL in the bioinformatics application of protein fold prediction, which is a key step towards understanding the function of proteins and thus crucial for drug design. We found that using non-sparse $\ell_{p=1.14}$-norm MKL significantly increases the predictive accuracy by over 5% points compared to ℓ_1-norm MKL, the SVM using a uniform kernel combination, and the SVM using the best kernel determined by model selection. The optimality of such an intermediate p can be explained by the presence of a large number of "weak" (but nevertheless complementary, informative) kernels.

5.4.3 Metabolic Network Reconstruction—A Uniform Scenario

Application description and goal A *metabolic network* is a schematic representation of the set of enzymes residing in a cell together with their functional interactions. As such it allows us to understand the connections between the genome and the physiology of an organism. We can think of a metabolic network as a graph where each enzyme is represented by a node and there is an edge between two enzymes if they interact (see example shown in the figure to the right). Examples of metabolic networks include glycolysis, Krebs cycle, and the pentose phosphate pathway. The task in metabolic network reconstruction is, given a partial network built from a subset of enzymes, to predict functional interactions of new enzymes.

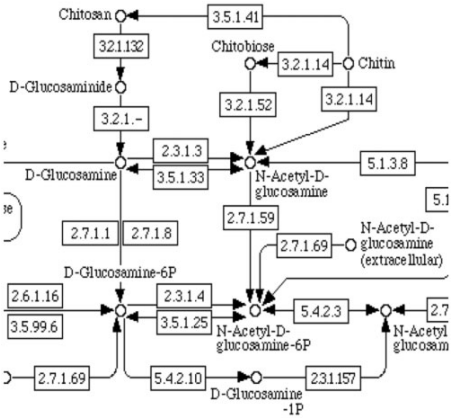


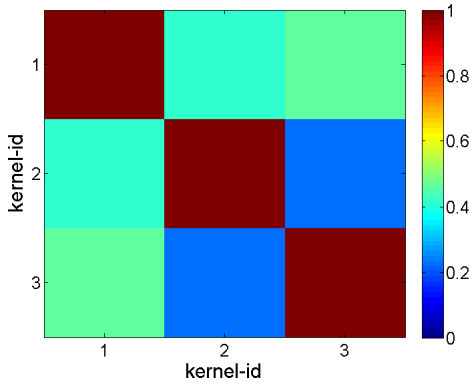
Figure 5.9: Part of an aminosugars metabolic network (figure taken from Bleakley et al., 2007).

There is a growing interest in this application within the bioinformatics community due to the accumulation of biological information about enzymes such as gene expression data, phylogenetic data, and location data of enzymes in the cell—in part this can be attributed to the sequencing of complete genomes. In the study of Bleakley et al. (2007), a new method to predict potentially new relationships was proposed and its effectiveness demonstrated. We recognize this method as a regular SVM using a uniform combination of kernels and thus are interested in studying whether ℓ_p-norm MKL can help in making even more accurate reconstructions.

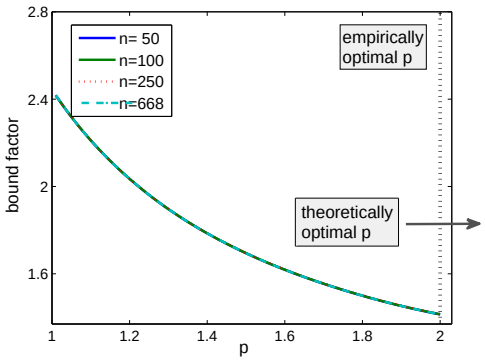
Experimental setup We use a data set that was originally studied in Yamanishi et al. (2005): it consists of 668 enzymes of the yeast *Saccharomyces cerevisiae* and 2782 functional relationships extracted from the KEGG database (Kanehisa et al., 2004). We employ the experimental setup of Bleakley et al. (2007), who phrase the task as graph-based edge prediction with local models by learning a model for each of the 668 enzymes. The provided kernel matrices capture expression data (EXP), cellular local-

	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 (SVM)	$\mathbf{82.94 \pm 0.8}$	$(\mathbf{82.1 \pm 2.2})$
1-norm MKL	74.7 ± 1.5	
1.33-norm MKL	81.30 ± 1.4	
1.85-norm MKL	82.90 ± 1.0	
2.28-norm MKL	82.85 ± 0.9	
2.66-norm MKL	82.24 ± 0.9	
4-norm MKL	80.37 ± 0.7	
Recombined and product kernels		
1-norm MKL	79.05 ± 0.5	
1.14-norm MKL	80.92 ± 0.6	
1.33-norm MKL	81.95 ± 0.6	
1.6-norm MKL	$\mathbf{83.13 \pm 0.6}$	

(a) Results (in AUC).



(b) Alignments



(c) Bound (curves are equall for all n)

Figure 5.10: The metabolic network reconstruction experiment: results and analysis.

ization (LOC), and the phylogenetic profile (PHY); in addition, 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; on average, we train 534 enzyme-based models in each fold; 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 5.10 (c) shows, this results in slightly better AUC values for single kernel SVMs; the results by Bleakley et al. (2007) are shown in brackets.

Results The results are shown in Figure 5.10 (a); the results by Bleakley et al. (2007) for single-kernel SVM are shown in brackets. We can observe from the figure that the SVM using a uniform kernel combination performs best, although its solution is well approximated by ℓ_p-norm MKL using values that are slightly smaller or larger than $p = 2$. Increasing the number of kernels by including recombined and product kernels does improve the results obtained by MKL, especially for small values of p, but even the highest AUC values are not significantly different from those of the uniform kernel combination.

Interpretation It suggests that the performance of the SVM using a uniform kernel combination can be explained by all three kernels performing well individually. Their correlation is only moderate, as shown in Figure 5.10 (c), indicating that they contain complementary information. Hence, downweighting one of those three kernels leads to a decrease in performance, as observed in our experiments. This explains why the uniform kernel combination yields the best prediction accuracy in this experiment.

Bound Figure 5.10 (c) shows bound factor as a function of p for various values of n. Note that regardless of n the curves fall into the same regions, rendering them impossible to differentiate in the plot. The optimal p in the interval $p \in [1, 2]$ is determined as $p = 2$, although the plot indicates that even a higher p could be optimal. In contrast to previous applications of the bound, we do not observe a trend in the bound when increasing n. Inspecting the $\boldsymbol{w}^{\mathrm{SVM}}$ as output by the best performing algorithm (namely, the SVM using a uniform kernel combination), we find $(\|\boldsymbol{w}_m^{\mathrm{SVM}}\|)_{m=1,2,3} = (0.332, 0.330, 0.338)$ for $n = 50$ and $(0.337, 0.326, 0.337)$ for the maximal $n = 668$. Clearly, there is hardly a development towards a more sparse w when increasing n so that it is not surprising that the bounds coincide with varying n.

Summary and discussion We studied ℓ_p-norm MKL in the bioinformatics application of metabolic network reconstruction. With the rapidly growing "industrialization" of sequencing technologies there is growing demand for the reconstruction of metabolic cycles such as the Krebs cycle. We found the uniform kernel combination working best in this application because, as determined by our kernel alignment technique, all three kernels encode strong and complementary features. The uniformness of this scenario can also be seen from the bounds, which indicate a higher p to be optimal than in the applications we have seen before.

5.5 Computer Vision Experiment: Object Recognition

Application description and goal In *object recognition* the task is to find an object in an image. Traditionally, this is easily handled by humans, even if the object is rotated or partially obstructed from view, but for machines it poses quite a challenge. This is because objects can be shown from various view points and thus can come in various shapes and scales.

A specialty of computer-based object recognition approaches is that a variety of feature descriptors are given, varying in, for example, color, texture, and shape information; each combination of those descriptors gives rise to a kernel. Cross-validation-based model selection is known to fail for this data as the descriptors capture *complementary* information (Gehler and Nowozin, 2009). Instead, researchers frequently resort to heuristics such as a uniform kernel combination. However, this does not account for varying relevances of the features that change across the tasks/concepts/classes. For example, color information increases the detection rates of stop signs substantially but is almost useless—or even counterproductive due to the introduction of additional noise—for finding cars. Clearly, this is because stop signs are colored red while cars can come in any color. As additional, non-essential features not only slow down computation but may also harm the predictive performance, it is necessary to combine the features in a more meaningful way to achieve state-of-the-art object recognition performance in real systems. In this thesis, we thus investigate whether ℓ_p-norm MKL can help in this task.

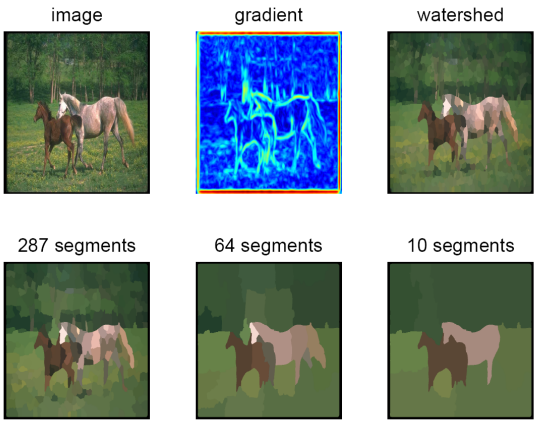
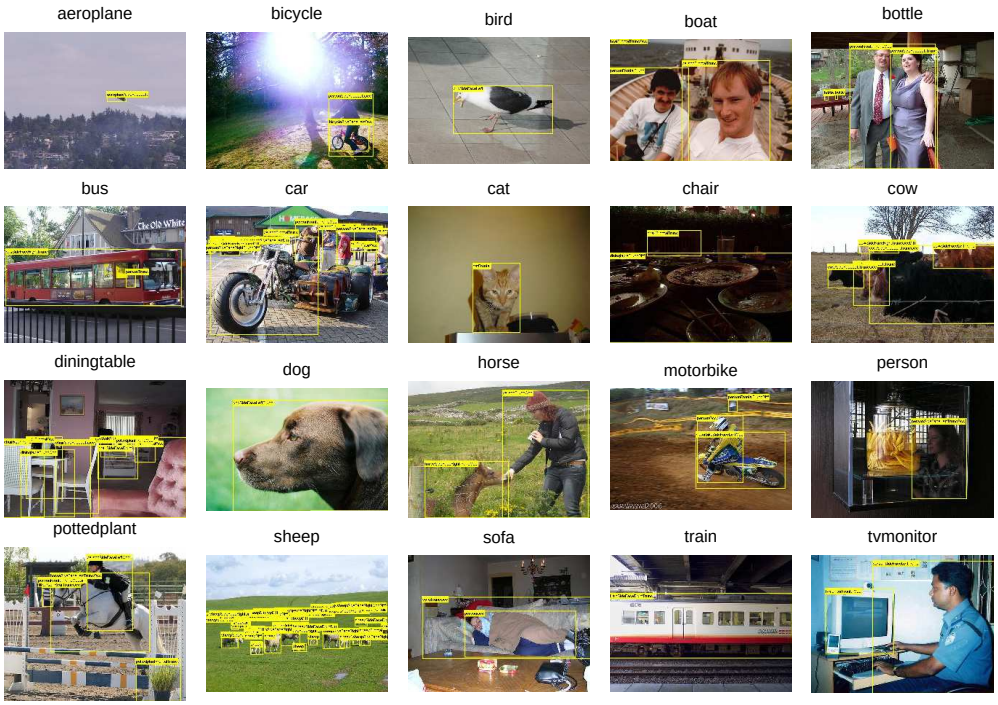


Figure 5.11: Figure taken Bach (2008a).

Data Set We experiment on the *PASCAL Visual Object Classes Challenge 2008* (VOC 2008) data (Everingham et al., 2008), which consists of 8780 images that are officially split by the organizers into 2113 training, 2227 validation, and 4340 test images. The organizers also provide annotation of the images in 20 object classes, where each image can be annotated by multiple objects (Figure 5.12 (a) shows examples of the occurring object classes). The official task posed by the challenge organizers consists in solving 20 binary classification problems, i.e., predicting whether at least one object from a class k is visible in a test image. The evaluation measure is the average precision (AP).

Feature Extraction We deploy 12 kernels that are inspired by the approaches of the winners of the VOC 2007 challenge (Marszalek and Schmid, 2007; Bosch et al., 2007). They are based on the following three types of features: Histogram of visual words (HoW), Histogram of oriented gradients (HoG), and Histogram of pixel colors (HoC).



(a) VOC 2008 DATA SET: Exemplary images. Note that an image can be annotated by multiple concepts (e.g., the above *horse* image is also annotated with *person*).

	average	aeroplane	bicycle	bird	boat	bottle
1-norm MKL	40.8±0.9	**66.9±6.8**	36.4±6.6	44.1±5.7	56.8±5.0	**19.2±3.8**
SVM	40.8±1.0	66.4±6.6	39.1±5.9	43.3±5.8	**57.5±5.0**	18.4±3.6
p-norm MKL	**42.3±0.9**	**67.1±6.3**	**40.7±6.6**	44.7±5.4	**57.8±5.4**	**19.5±3.6**
selected p		1.07	1.06	1.11	1.11	1.11
		bus	car	cat	chair	cow
1-norm MKL		39.3±10.6	49.0±2.8	47.7±3.8	**44.1±4.9**	**10.8±3.5**
SVM		**42.3±9.1**	48.9±3.3	46.1±3.2	43.0±4.7	**8.2±2.9**
p-norm MKL		**41.7±9.5**	**50.3±3.4**	**48.9±3.7**	**44.9±3.8**	**10.3±3.1**
selected p		1.33	1.11	1.0588	1.0154	1.0448
		diningtable	dog	horse	motorbike	person
1-norm		27.1±7.0	**34.4±4.4**	39.6±5.8	41.7±4.5	84.1±1.3
SVM		**29.5±9.1**	**33.2±2.7**	**42.5±6.5**	42.8±2.9	83.9±1.2
p-norm MKL		**30.1±6.2**	**34.0±3.4**	**42.0±6.6**	**44.7±4.2**	**84.5±1.2**
selected p		1.1351	1.1351	1.1579	1.1111	1.0588
		pottedplant	sheep	sofa	train	tvmonitor
1-norm		**14.7±4.5**	**26.3±7.7**	**33.0±7.2**	50.9±9.8	50.8±5.5
SVM		**15.5±4.3**	22.9±7.0	31.3±6.5	50.9±9.2	**51.0±5.7**
p-norm MKL		**16.1±4.7**	**27.5±7.6**	**33.9±6.9**	**53.7±9.9**	**52.9±5.7**
optimal p		1.1111	1.0154	1.1351	1.1111	1.0588

(b) Results (in AP)

Figure 5.12: The computer vision experiment: data set and results.

1. The HoW features are constructed in a standard way as in Csurka et al. (2004): first, the SIFT descriptors (Lowe, 2004) are calculated on a grid of 10 pixel pitches for each image using a code book of size 1200, which is learned by k-means clustering deploying the SIFT descriptors extracted from the training images. Then, all SIFT descriptors are assigned to visual words (prototypes) and summarized into histograms within entire images or sub-regions, deploying HoW features over the grey and hue channels.

2. The HoG features over the grey channel are computed by discretizing the orientation of the gradient vector at each pixel into 40 bins and by summarizing the discretized orientations into histograms within image regions (Dalal and Triggs, 2005; Bosch et al., 2007). Canny detectors (Canny, 1986) are used to discard contributions from pixels, around which the image is almost uniform.

3. The HoC features over the hue channel are constructed by pixel-wise discretizing color values and computing their histograms within image regions.

The above features are histograms containing no spatial information. We therefore enrich the respective representations by a pyramidal approach (Lazebnik et al., 2006) to also capture the spatial context of an image. Furthermore, we apply a χ^2 kernel (Zhang et al., 2007) on top of the enriched histogram features, where the bandwidth of the χ^2 kernel is heuristically chosen (proportional to the mean of the squared Euclidean distance averaged over all training example pairs (Lampert and Blaschko, 2008)). In total, we prepared 12 kernels by mutually combining 4 feature types/colors $\{\mathrm{HoW}_g$, HoW_h, HoG_g, $\mathrm{HoC}_h\}$ with 3 pyramid levels. All kernels were spherically normalized according to Eq. (3.7).

Experimental setup The original data set contains a test set, but with the labels being undisclosed. We therefore create 10 random splits of the unified training and validation sets into new, smaller sets containing 2111 training, 1111 validation, and 1110 test images. For each of the 10 splits, the training images are used for learning classifiers, while the SVM regularization parameter C and the MKL norm parameter p are chosen based on the maximal AP score on the validation images. Thereby, the regularization constant C and the norm parameter p are optimized class-wise from the candidates $\{2^{-4}, 2^{-3}, \ldots, 2^4\}$.

For a thorough evaluation, we would have to construct another codebook for each cross-validation split. However, creating codebooks and assigning descriptors to visual words is a time-consuming process. In our experiments, we therefore resort to the common practice of using a single codebook created from all training images contained in the official split. Although this could result in a slight overestimation of the AP scores, this affects all methods equally and does not favor any classification method over another—our focus lies on a *relative* comparison of the different classification methods; therefore there is no loss in exploiting this computational shortcut.

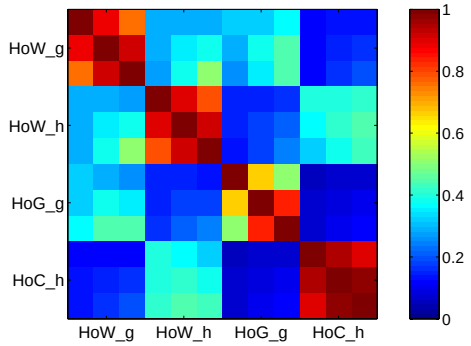
Note that we exclude a comparison to single-kernel SVMs since it is clear for this application that a single kernel alone cannot capture the relevant information needed for this task (see, e.g., Gehler and Nowozin, 2009, and references therein).

Results The results are shown in Table 5.12 (b). Boldface shows the best method as well as all other ones that are not statistical-significantly worse according to a Wilcoxon signed-ranks test at a significance level of 5%. Medians of the optimal norm parameter p are shown. One can observe that $\ell_{p>1}$-norm MKL is among the best algorithms independent of the concept class, except for *bird* where it attains the maximal AP score, but insignificantly so. For many concepts, the AP score of $\ell_{p>1}$-norm MKL is considerably higher than the one attained by 1-norm MKL or the SVM using the uniform kernel combination: e.g., *bicycle*, *car*, *cat*, *motorbike*, and *train* are classes where learning kernels substantially helps. On average, the AP scores increases by about 1.5 AP Points by using non-sparse $\ell_{p>1}$-norm MKL. It is also interesting to note, that when comparing ℓ_1-norm MKL with the SVM using a uniform kernel combination, there is no consistent picture over the classes: while ℓ_1-norm MKL is better for the *aeroplane*, *bottle*, *chair*, *sheep* and *sofa* classes, it is outperformed by the regular sum-kernel SVM for *boat*, *bus*, *diningtable*, *horse* and *tvmonitor*. By optimizing the non-sparsity parameter p in between the two extreme cases, our procedures can always outperform both baselines.

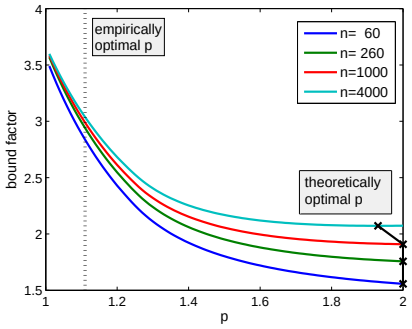
Interpretation We now analyze the kernel mixtures $\boldsymbol{\theta}$ as output by MKL. To this aim, we first compute the pairwise kernel alignment scores of the 12 base kernels (shown in Figure 5.13 (a)). One can see from the figure that the tree kernels constructed from the same type/color features have high similarities, as easily identified from the four diagonal blocks. Across different features, we observe mid-range similarities between those with the same type (HoW_g – HoW_h) and with the same colors (HoW_g – HoG_g, HoW_h – HoC_h). As expected, the three different pyramid levels of each type/color feature are very closely aligned. The pair of the HoW features are indeed located closer to each other than to the other features.

Intuitively, the kernel weights matter only moderately when the base kernels are similar. This situation holds for the three pyramid levels of each type/color feature. Therefore, we also try combining these three kernels with uniform weights in advance and optimized the four weights corresponding to feature types/colors afterwards. In fact, this pre-combination procedure does not degrade the classification performance: its mean AP is 42.3 ± 0.7. Especially, when further increasing the number of employed kernel parameters, this pre-averaging procedure becomes particularly appealing.

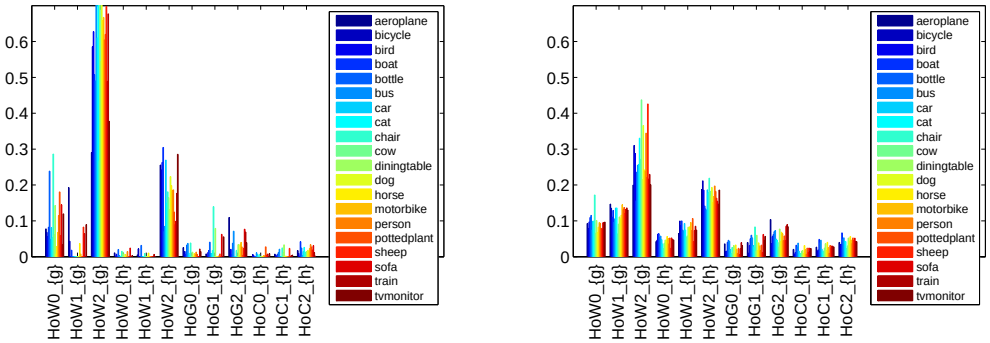
Based on the properties of the 12 base kernels, we check their kernel weights as output by 1-norm and p-norm MKL. Figure 5.13 (c) shows the average kernel weights over 10 repetitions of the experiment. We see that ℓ_1-norm MKL focuses on only a few informative kernels such as $\mathrm{HoW2}_g$, $\mathrm{HoW0}_g$ and $\mathrm{HoW2}_h$, almost completely neglecting HoG_g and HoC_h (Left). By contrast, the solutions output by $\ell_{p>1}$-norm MKL (Right) are more balanced: while the HoW kernels get relatively high weights, the HoG and HoC features get considerably lower weights but are not ignored completely. We can also observe that ℓ_1-norm MKL tends to completely discard color information while it is still incorporated into the final model by $\ell_{p>1}$-norm MKL. But color information can be crucial for object detection: for example, school buses tend to be yellow and horses are more frequently found on green lawns than on gray roads; discarding the color



(a) Alignments



(b) Bound



(c) MKL weights $\boldsymbol{\theta}$: ℓ_1-norm (left) and ℓ_p-norm MKL (right).

(d) Examples: Images where MKL helped performance most. *Left: a bus ranked 39th (ℓ_1-norm) and 6th (ℓ_p-norm MKL), respectively. Right: a horse ranked as 142th (ℓ_1-norm) and as 41st (ℓ_p-norm MKL).*

Figure 5.13: The computer vision experiment: interpretation and analysis.

information, as done by ℓ_1-norm MKL, hurts performance on concept classes that are well characterized by colors—this might be one of the major reasons for the superior performance of ℓ_p-norm MKL on these images compared to the ℓ_1-norm MKL.

Figure 5.13 (d) shows example images where ℓ_p-norm MKL greatly helps generalization performance. In both cases, ℓ_1-norm MKL tends to focus on the finest pyramid layer, and tries to capture layout information. However, since the VOC dataset includes a substantial number of not-so-well-aligned images (such as the examples shown in the figure), focusing on layout might sometimes fail. In contrast, ℓ_p-norm MKL distributes more weights to coarser pyramid layers, and thus successfully identifies these images.

Bound From the bound shown in Figure 5.13 (b), we observe an overestimation of the optimal p. The empirical best p was $p = 1.11$ in median. In contrast, the bound predicts a p with $p \approx 2$. This is very similar to what be have already observed in the protein fold prediction experiment, presented in Section 5.4.2. Interestingly, both experiments, the present one and the protein folding experiment, consist of multi-label classification tasks with 20 and 27 classes, respectively. This indicates that, for small n, the bounds are not reliable in multi-label classification problems. Nevertheless, we can observe a trend towards a more accurate prediction of the optimal p with increasing n, which is similar to the previous experiments. It could thus be conjectured that with increasing n the bound's prediction will eventually coincide with what is also measured empirically.

Summary and discussion In this chapter, we applied ℓ_p-norm MKL to object recognition in computer vision. While humans can easily recognize objects in images, even if they are rotated or partly covered, machines frequently fail in this task. Since images are characterized by heterogeneous kernels arising from color, texture, or shape information, but the usefulness of a particular kernel strongly depends on the object to be discovered (for example, cows are frequently found in front of green lawns while cars can come in any color), it is very appealing to use MKL in this application. Indeed, in our experiments on the official VOC08 challenge data set, we observed a substantial improvement in using ℓ_p-norm MKL—this holds over the whole range of object classes. Using the alignment technique we connected this fact with the color information being discarded by ℓ_1-norm MKL while it is overly incorporated into the uniform kernel combination. This is especially remarkable since the uniform SVM was shown to be a tough competitor in the past (Gehler and Nowozin, 2009). The latter might also be connected with prior knowledge that is incorporated by researchers when the set of kernels is chosen a priori.

5.6 Summary and Discussion

We empirically analyzed the generalization performance of ℓ_p-norm MKL in terms of the test error and application-specific evaluation measures, respectively. To this end, we measured the accuracy of ℓ_p-norm MKL and compared it to the one achieved by the SVM baselines, using either a uniform kernel combination or a single kernel tuned by model selection. To analyze the results we developed a methodology based on

theoretical bounds (investigating the theoretically optimal p) and kernel alignments (analyzing the complementariness of the kernel set). We exemplified the methodology by means of a toy experiment, where we constructed data by gradually increasing the true underlying sparsity (as measured by the size of the sparsity pattern of the Bayes hypothesis), resulting in several constructed learning scenario that differ in their underlying sparsity. First, we observed the theoretical bounds reflecting the empirically observed optimal p in the idealized setup where we know the Bayes hypothesis; second, we showed that kernel alignments reflect the dependancy structure of the kernel set; third, we found that the choice of the norm parameter p can be crucial to obtain a high test set accuracy, concluding that in the presence of many irrelevant kernels, a small p is expected to be beneficial and an intermediate/large p will prove to be beneficial otherwise.

We then applied ℓ_p-norm MKL to diverse, highly topical and challenging problems from the domains of bioinformatics and computer vision on small and large scales. Data frequently arises in those domains from multiple heterogeneous sources or is represented by various complementary views, the right combination of which being unknown, rendering them especially appealing for the use of MKL. Beside investigating the effectiveness of ℓ_p-norm MKL in terms of test set accuracy, we were also interested in *understanding* the results, implementing the proposed methodology.

We exemplarily carried out the proposed experimental methodology for transcription splice site detection, a cutting-edge bioinformatics application. The data set we experimented on was also used in the previous study of Sonnenburg et al. (2006a) and comprises 93,500 instances. We showed that regardless of the sample size, ℓ_p-norm MKL can improve the prediction accuracy over ℓ_1-norm MKL and the SVM using a uniform kernel combination (which itself outperforms the single-kernel SVM using the kernel determined by model selection). This is remarkable and of significance for the application domain: the method using a uniform kernel combination was recently confirmed to be the most accurate TSS detector in a comparison of 19 state-of-the-art models Abeel et al. (2009). Furthermore, we observed that the choice of n had no impact on the empirical optimality of a particular p. We also evaluated the theoretical bounds and found that with increasing n they give a more and more accurate estimation of the optimal p through an increasingly tight approximation of the Bayes hypothesis.

Moreover, we also studied other applications from the domain of bioinformatics: protein subcellular localization, protein fold prediction, and metabolic network reconstruction. In a nutshell, we found this domain being too diverse to be categorized into a single learning scenario (e.g., sparse). In fact, we presented diverse bioinformatics applications covering the whole range of scenarios: sparse, intermediate non-sparse, and uniform. However, by appropriately tuning the norm parameter, ℓ_p-norm MKL proved robust in all scenarios.

Finally, we applied ℓ_p-norm MKL to object recognition, a burning topic in computer vision. The same setting was also studied earlier by Varma and Ray (2007) and Gehler and Nowozin (2009), who found classical MKL to not increase the prediction accuracy compared to the uniform kernel SVM baseline. We applied the developed ℓ_p-

norm MKL to this application domain and showed that, by taking $p = 1.11$ in median, the prediction accuracy could be raised over the SVM baselines, regardless of the object class, by an AP score of 1.5 in average, and for 7 out of the 20 classes significantly so.

Conclusion and Outlook

To successfully cope with the most recent challenges imposed by application domains such as bioinformatics and computer vision, machine learning systems need to effectively deal with the multiple data representations—or kernels—naturally arising in those domains. Previous approaches to learning with multiple kernels restrict the search space by convex combinations, promoting sparse solutions to support interpretability and scalability. Unfortunately, this is often too restrictive and hinders MKL being effective in practice, as demonstrated in a variety of recent works (e.g. Cortes et al., 2008; Gehler and Nowozin, 2008; Kloft et al., 2008b; Noble, 2008; Cortes, 2009; Gehler and Nowozin, 2009; Kloft et al., 2009a).

In this thesis, we addressed the problem of learning with multiple kernels by proposing the *ℓ_p-norm multiple kernel learning* methodology, deduced from a rigorous mathematical framework for learning with multiple kernels, unifying previous approaches to learning with multiple kernels under a common umbrella. In an extensive study, we evaluated ℓ_p-norm multiple kernel learning empirically in controlled environments as well as in highly topical real-work applications from the domains of computer vision and bioinformatics, comparing our methodology to even its strongest competitors such as the support vector machine using a uniform kernel combination. We found ℓ_p-norm multiple kernel learning being more efficient and more accurate than previous approaches to learning with multiple kernels, allowing us to deal with up to ten thousands of data points and thousands of kernels at the same time, as demonstrated on several data sets.

Our methodology is underpinned by deep foundations in the theory of learning: we proved tight lower and upper bounds on the complexity of the hypothesis class associated with ℓ_p-norm multiple kernel learning, giving theoretical guarantees on the fit of our method and showing that, depending on the true underlying learning task, our framework can attain stronger theoretical guarantees than classical approaches to learning with multiple kernels. Stimulated by recent attempts to catalyze research in the direction of *understanding* learning with multiple kernels (Lanckriet et al., 2009), we showed that the optimality of ℓ_p-norm multiple kernel learning can be connected with the stronger theoretical guarantees that it attains in practice, compared to classical approaches.

The proposed framework does not generally require the incorporation of expert knowledge; however, such application-specific knowledge can be used to further improve its performance, for example, by pre-selecting or upscaling kernels with high information content. In combination with this technique, ℓ_p-norm multiple kernel learning can even cope with the most challenging problems posed by application domains.

Summary of Results

The main results of this thesis can be summarized as follows.

Chapter 2. We presented a mathematical framework for learning with multiple kernels, comprising most existing lines of research in that area and rigorously generalizing the classic approach of Bach et al. (2004). We unified previous formulations under a common umbrella, which allowed us to analyze a large variety of MKL methods jointly, as exemplified by deriving a general dual representation of the criterion, without making assumptions on the employed norms and losses, beside the latter being convex. This delivered insights into connections between existing MKL formulations and, even more importantly, can be used to derive *novel* MKL formulations as special cases of the framework.

Chapter 3. We derived efficient algorithms to solve the ℓ_p-norm MKL optimization problem. All algorithms, for the sake of performance, are implemented in C++, but for usability, are equipped with interfaces to MATLAB, Octave, Phyton, and R. We also wrote macro scripts completely automating the whole process from training over model selection to evaluation. Our software is freely available under the GPL license. In our computational experiments, we found these large-scale optimization algorithms allowing us to deal with ten thousands of data points and thousands of kernels at the same time, as demonstrated on the MNIST data set. We compared our algorithms to the state-of-the-art in MKL research, namely HessianMKL (Chapelle and Rakotomamonjy, 2008) and SimpleMKL (Rakotomamonjy et al., 2008), and found ours to be up to two magnitudes faster. We proved the convergence of our AnalyticalMKL using the usual regularity assumptions.

Chapter 4. The proposed techniques are also theoretically founded: we proved tight lower and upper bounds on the local and global Rademacher complexities of the hypothesis class associated with ℓ_p-norm MKL, from which we derived excess risk bounds with fast convergence rates, thus being tighter than existing bounds for ℓ_p-norm MKL (Cortes et al., 2010a), which only achieved slow convergence rates. While for our results to hold we employed an assumption on the independence of the kernels, related works on sparse recovery required similar assumptions (Raskutti et al., 2010; Koltchinskii and Yuan, 2010). We connected the minimal values of the bounds with the structured soft sparsity of the underlying Bayes hypothesis, demonstrating that for a large range of learning scenarios ℓ_p-norm MKL attains substantially stronger generalization guarantees than classical MKL, justifying the use of ℓ_p-norm MKL and multiple kernel learning in general.

Chapter 5. In an extensive evaluation, we empirically measured the generalization performance of ℓ_p-norm MKL on diverse and relevant real-world applications from bioinformatics and computer vision. We developed a methodology based on the theoretical bounds and kernel alignments for analyzing the results, as exemplified by means of a toy experiment, where we observed the theoretical bounds and the kernel alignments well reflecting the empirically observed optimal p. We carried out the experimental methodology for genomic transcription splice site detection. We showed that regard-

less of the sample size, by taking $p = 4/3$, ℓ_p-norm MKL can improve the prediction accuracy over ℓ_1-norm MKL and SVM using a uniform kernel combination, which was recently confirmed to be the most accurate TSS detector in an international comparison of 19 models Abeel et al. (2009). We also evaluated the theoretical bounds and found them with increasing n increasingly accurate reflecting the optimal p. We applied ℓ_p-norm MKL to protein fold prediction, achieving an significant improvement of 6% in accuracy. Finally, we applied ℓ_p-norm MKL to object recognition, a burning topic in computer vision, studied earlier by Varma and Ray (2007) and Gehler and Nowozin (2009) who found no improvement in using classical MKL. In contrast, we showed that, by taking $p = 1.11$ in median, ℓ_p-norm MKL outperforms even the best SVM competitors, regardless of the object class.

Future Work

As recently argued in Kloft et al. (2010b), among the most important challenges in multiple kernel learning is the exploration of new objectives and parameterizations. To this end, an interesting approach was recently undertaken by Yan et al. (2010), who studied a ℓ_2-norm MKL variant based on a Fisher discriminant objective function with very promising results on object categorization tasks. Similar objectives were also explored in Yu et al. (2010), who presented a more general formulation. First steps towards new ways of combining kernels were undertaken in Varma and Babu (2009) and Cortes et al. (2009b), respectively, where non-linear combinations (products and polynomials, respectively) were studied. Unfortunately, in the current state of research, both formulations in general lead to non-convex optimization problems and the corresponding objectives are hard to optimize accurately.

Besides classification, which forms the core of machine learning and on which we focused in this thesis, it is interesting to study the proposed methodology also in other learning tasks such as regression or novelty detection tasks. A first step in this direction was undertaken in Cortes et al. (2009a), where ℓ_p-norm MKL was applied to regression tasks on Reuters and various sentiment analysis datasets, but restricted to the case $p = 4/3$. Here it would be interesting to exploit the full power of our framework by studying the whole range of $p \in [1, \infty]$. Another approach towards different learning tasks was undertaken in Kloft et al. (2009b), where density-level set estimation and novelty detection were studied with promising first results. Likewise, semi-supervised learning (Chapelle et al., 2006) is to be explored. Also, online learning recently gained considerable attention (Orabona et al., 2010) and is subject to current work (Martins et al., 2011; Orabona and Jie, 2011).

Future work could also aim at further exploring applications of ℓ_p-norm MKL. Ongoing work in the field of computer vision is based on Nakajima et al. (2009b), investigating ℓ_p-norm multiple kernel learning on photo annotation tasks with promising first results. Also, effectively learning taxonomies (Binder et al., 2011) has recently gained attention from the scientific community and could benefit from the use of MKL. Also, in the bioinformatics domain, we observed applications of ℓ_p-norm MKL, most

notably, the study by Yu et al. (2010) on two real-world genomic data sets for clinical decision support in cancer diagnosis and disease relevant gene prioritization, respectively, showing a substantial improvement in using $\ell_{4/3}$-norm MKL compared to the SVM baselines. Again, it would be interesting to refine the analysis by searching for an optimal p and, moreover, evaluate our bounds and kernel alignments. Another interesting application is computer security, where MKL was successfully applied to network intrusion detection tasks (Kloft et al., 2008a), using appropriate loss functions.

It would also be interesting to further theoretically explore the ℓ_p-norm multiple kernel learning framework, for example, beyond the traditional borders of the local Rademacher complexities. Clearly, future work here could built upon Bartlett and Mendelson (2006). Another theoretical challenge is to analyze more complex hypothesis classes. The unifying framework presented in this thesis might serve as a good starting point for such an undertaking, as exemplified for the case of generalized elastic-net-style classes in Kloft et al. (2010a). This is becoming especially pressing with respect to non-isotropic hypothesis classes that might also be profitable in practice since they allow for the incorporation of expert knowledge. A considerable amount of work on those non-standard classes would also have to focus on optimization algorithms.

Finally, an innovative approach was recently taken in Widmer et al. (2010), where $\ell_{p>1}$-norm MKL was exploited as a methodical tool to learn hierarchy structures in multi-task learning, thus extending the work of Evgeniou and Pontil (2004) to the structured domain. This indicates the many possibilities for further exploration of our framework beyond the standard setting, for example, exploring the relations to learning distance metrics or learning the covariance function of a Gaussian process—clearly, this requires research that goes beyond the classical setup.

Appendix

A Foundations

In this appendix, we introduce the foundations of this thesis; the presentation in part follows the textbooks of Shawe-Taylor and Cristianini (2004) and Boyd and Vandenberghe (2004). We introduce kernel methods, Rademacher theory, and Lagrangian duality.

A.1 Why Using Kernels?

Consider a typical machine learning problem. Let us assume, we are given a set of images and the task is to build a machine that categorizes the images of 256×256 pixels with respect to whether they contain a certain object, for example, a vehicle (this models a scenario where the images are obtained from the visual perception module of a driverless vehicle and the goal is to evade other vehicles). One possible method to obtain features from such an image could be through considering the gray value of each pixel. This way we can represent each image by a 256^2-dimensional vector. We thus aim at finding a partition of $\mathbb{R}^{256\times256}$ into two sets: the one that consists of the points representing images that are likely to contain cars and the one that is likely to *not* contain cars.

As it turns out, this problem is to hard to be solved by a simple, linear partition of the space. *Kernel methods* offer an effective and computationally efficient solution to this problem. Kernel methods can be understood as the following scheme (Shawe-Taylor and Cristianini, 2004, p. 26):

1. Data items are embedded into a vector space called feature space.

2. A hyperplane separating the images of the data items in the feature space is sought.

3. The algorithms are implemented in such a way that the coordinates of the embedded points are not needed, only their pairwise scalar products.

4. The pairwise scalar products can be efficiently computed directly from the original data items using a kernel function.

The above scheme is illustrated in Figure A.1–A.2. Formally, we define a kernel as follows:

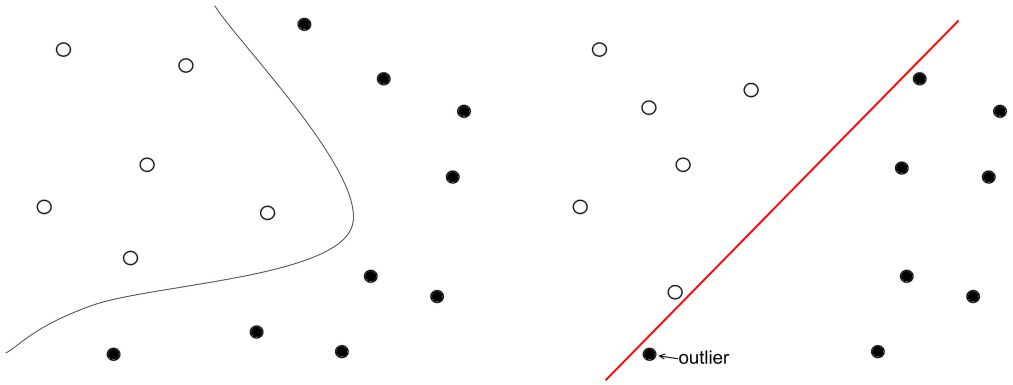


Figure A.1: Illustration of kernel methods: linear separation is not always possible (LEFT) or desired (RIGHT).

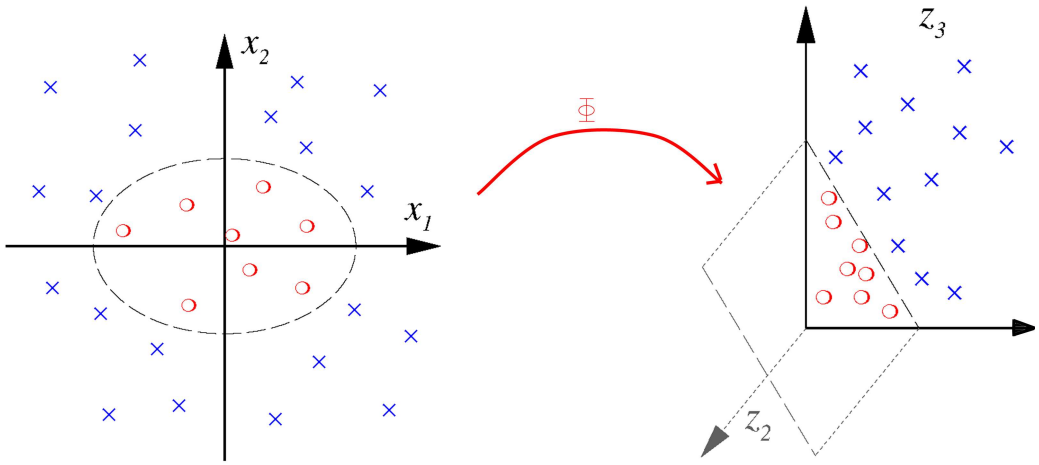


Figure A.2: Kernel methods address this by mapping the data into a higher dimensional space to allow for linear separation. This a modification of a figure taken from Müller et al. (2001).

Definition A.1 (Kernel)**.** *Given an input space $\mathcal{X}$, a* kernel *is a function* $\mathrm{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *such that there is a (Hilbert) feature space $\mathcal{H}$ and a mapping*

$$\phi : x \longmapsto \phi(x) \in \mathcal{H}$$

so that for all $x, \bar{x}$

$$k(x, \bar{x}) = \langle \phi(x), \phi(\bar{x}) \rangle .$$

The definition of a kernel uses the notion of a Hilbert space, which can be thought of as a continuous generalization of a Euclidean vector space and is formally defined as a vector space together with a scalar product that is complete and separable. Completeness refers in this case to every Cauchy sequence converging in $\mathcal{H}$ and separability means that $\mathcal{H}$ admits a countable orthonormal basis.

How can we check whether a function k is indeed a kernel? If we know that k is a continuous function, then there is a nice characterization: given a sample $\{x_1, \ldots, x_n\}$, the matrix $K = \big(k(x_i, x_j)\big)_{i,j=1}^{n}$ formed from the pairwise kernel evaluations is called *kernel matrix*; one can show that k is a kernel if and only if for any given sample the kernel matrix K is positive semi-definite.

A.2 Basic Learning Theory

Empirical risk minimization Arguably the most important property of successful learning machines is that they find decision functions which not only characterize the training data $(x_1, y_1), \ldots, (x_n, y_n) \subset \mathcal{X} \times \mathcal{Y}$ well but also characterize new and unseen data. Many such algorithms underlie the concept of *empirical risk minimization* (ERM), that is, for a given hypothesis class $H \subset \mathcal{Y}^{\mathcal{X}}$ a minimizer

$$f^* \in \operatorname{argmin}_{f \in H} \, L_n(f)$$

is searched for, where $L_n(f) = \sum_{i=1}^{n} l\left(f(x_i), y_i\right)$ is the (cumulative) *empirical loss* of a hypothesis f with respect to a function $l : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$ (called *loss function*). The name empirical risk minimization stems from the fact that $L_n(f)$ being n times the empirical risk of f. Kernel approaches consider linear models of the form

$$f_{\boldsymbol{w}}(x) = \langle \boldsymbol{w}, \phi(x) \rangle$$

together with a (possibly non-linear) mapping $\phi : \mathcal{X} \to \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space. Often the hypothesis class is isotropically parameterized by a parameter D:

$$H_D = \left\{ f_{\boldsymbol{w}}(x) \,\middle|\, \|\boldsymbol{w}\|_2 \leq D \right\}$$

(denoting by $\|\boldsymbol{w}\|_2$ the Hilbert-Schmidt norm in $\mathcal{H}$), which allows to "kernelize" (Schölkopf et al., 1998) the resulting models and algorithms, that is, formulating them solely in terms of inner products $k(x, x') := \langle \phi(x), \phi(x') \rangle$ in $\mathcal{H}$.

Rademacher theory A useful quantity to theoretically analyze ERM is the Rademacher complexity; let $x_1, \ldots, x_n \subset \mathcal{X}$ be an i.i.d. sample drawn from a probability distribution P; then, the global Rademacher complexity of a hypothesis class H is defined as

$$R(H) = \mathbb{E} \sup_{f_{\boldsymbol{w}} \in H} \left\langle \boldsymbol{w}, \frac{1}{n} \sum_{i=1}^{n} \sigma_i \phi(x_i) \right\rangle ,$$

where $(\sigma_i)_{1 \leq i \leq n}$ is an i.i.d. family (independent of $\phi(x_i)$) of Rademacher variables (random signs). Its empirical counterpart is denoted by

$$\widehat{R}(H) = \mathbb{E}\big[R(H) \big| x_1, \ldots, x_n\big] = \mathbb{E}_{\boldsymbol{\sigma}} \sup_{f_{\boldsymbol{w}} \in H} \left\langle \boldsymbol{w}, \frac{1}{n} \sum_{i=1}^{n} \sigma_i \phi(x_i) \right\rangle .$$

The interest in the global Rademacher complexity comes from the fact that, if known, it can be used to bound the generalization error (Koltchinskii, 2001; Bartlett and Mendelson, 2002):

Theorem A.2 (Bartlett and Mendelson, 2002, Theorems 8 and 12.4)**.** *Let $l : \mathbb{R} \supseteq \mathcal{Y} \to [0, 1]$ be a* L*-Lipschitz continuous loss function. With probability larger than $1 - \delta$ it holds uniformly for all classifiers $f \in H$*

$$\mathbb{E}[l(yf(x))] \leq \frac{1}{n} \sum_{i=1}^{n} l(y_i f(x_i)) + 2LR(H) + \sqrt{\frac{8 \ln \frac{2}{\delta}}{n}} .$$

The above theorem says that in order to obtain a generalization bound for an ERM-based learner it suffices to bound the global Rademacher complexity. We illustrate this for the support vector machine (SVM) as an example.

Example (SVM) An illustrative example for such a learning algorithm is the support vector machine which finds a linear function (eventually in a higher-dimensional feature space by using a kernel function) with an additional stability property: having a (soft) margin to a predefined fraction of training data; this property allows to well deal with outliers in the data (see Figure A.3). Formally the SVM is defined as follows:

$$\begin{aligned} \inf_{\boldsymbol{w}, \boldsymbol{t}} \quad & \frac{1}{n} \sum_{i=1}^{n} \max\left(0, 1 - y_i \left\langle \boldsymbol{w}, \phi(x_i) \right\rangle\right) \\ \text{s.t.} \quad & \|\boldsymbol{w}\|_2 \leq D . \end{aligned} \tag{SVM}$$

We recognize the SVM as an ERM algorithm using the hypothesis class $H = \big\{f : x \mapsto \langle \boldsymbol{w}, \phi(x) \rangle \mid \|\boldsymbol{w}\| \leq D\big\}$ and loss function $l(t) = \max(0, 1 - t)$; however, for the theoretical analysis it is more convenient to consider the truncated loss class

$$l_{\text{SVM}}(t) = \min(1, \max(0, 1 - t)) .$$

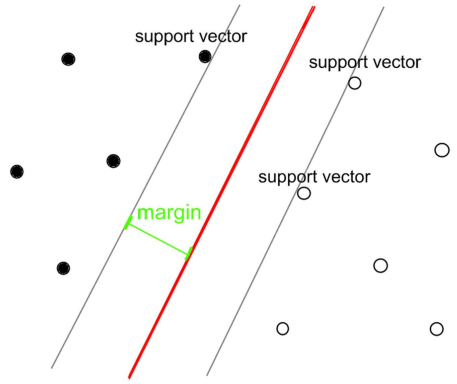


Figure A.3: Illustration of support vector machine: finding a linear hyperplane with maximum margin.

Subsequently, we can use the Cauchy-Schwarz inequality to bound the global Rademacher complexity of H as follows:

$$\begin{aligned}
R(H) &= \mathbb{E} \sup_{f_{\boldsymbol{w}} \in H} \left\langle \boldsymbol{w}, \frac{1}{n} \sum_{i=1}^{n} \sigma_i \phi(x_i) \right\rangle \\
&\overset{\text{C.-S.}}{\leq} \mathbb{E} \sup_{f_{\boldsymbol{w}} \in H} \|\boldsymbol{w}\| \left\| \frac{1}{n} \sum_{i=1}^{n} \sigma_i \phi(x_i) \right\| \\
&= \mathbb{E}\, D \left\| \frac{1}{n} \sum_{i=1}^{n} \sigma_i \phi(x_i) \right\| \\
&\overset{\text{Jensen}}{\leq} \mathbb{E}\, D \sqrt{\mathbb{E}_{\boldsymbol{\sigma}} \frac{1}{n^2} \sum_{i,j=1}^{n} \sigma_i \sigma_j k(x_i, x_j)} \\
&\overset{\sigma_i \text{ i.i.d.}}{=} \mathbb{E}\, D \sqrt{\frac{1}{n^2} \sum_{i=1}^{n} k(x_i, x_i)}
\end{aligned}$$

If we impose additional assumptions, for example, on the boundedness of the kernel, or boundedness of its moments, then we can use the above Rademacher bound together with the previous theorem, directly leading to a generalization error bound. For example, if the kernel is uniformly bound $\|k\|_\infty \leq B$, then we have

$$\mathbb{E}[l_{\text{SVM}}(yf(x))] \leq \frac{1}{n} \sum_{i=1}^{n} l_{\text{SVM}}(y_i f(x_i)) + 2D\sqrt{\frac{B}{n}} + \sqrt{\frac{8 \ln \frac{2}{\delta}}{n}}$$

because l_{SVM} is 1-Lipschitz.

A.3 Convex Optimization

A *convex optimization problem* (Boyd and Vandenberghe, 2004) is one of the form

$$\begin{aligned} \min_{\boldsymbol{v}\in\mathbb{R}^n} \quad & f(\boldsymbol{v}) \\ \text{s.t.} \quad & f_i(\boldsymbol{v}) \leq 0, \quad i = 1,\dots,m \\ & h_i(\boldsymbol{v}) = 0, \quad i = 1,\dots,l, \end{aligned} \tag{A.1}$$

where the functions $h_1,\dots,h_l : \mathbb{R}^n \mapsto \mathbb{R}$ are affine linear and $f_0,\dots,f_m : \mathbb{R}^n \mapsto \mathbb{R}$ are convex, i.e., satisfy $f_i(\alpha\boldsymbol{v} + \beta\tilde{\boldsymbol{v}}) \leq \alpha f_i(\boldsymbol{v}) + \beta f_i(\tilde{\boldsymbol{v}})$ for all $\boldsymbol{v},\tilde{\boldsymbol{v}} \in \mathbb{R}^n$ and all $\alpha,\beta \in \mathbb{R}$ with $\alpha,\beta = 1$, $\alpha \geq 0$, and $\beta \geq 0$.

There is no general analytical formula for the solution of such an optimization problem, but there are very effective methods for solving them, for example, interior-point methods and limited-memory quasi-Newton methods (Nocedal and Wright, 2006). If we can formulate a problem as a convex optimization problem, then we have solved the problem as we can usually solve it efficiently. In contrast, general, non-linear optimization problems can be very challenging, even with ten variables and intractable with a few hundreds of variables (Boyd and Vandenberghe, 2004, p. 8–9).

Lagrangian duality An important concept in optimization theory is the Lagrangian duality. The basic idea is to take the constraints in (A.1) into account by augmenting the objective function with a weighted sum of the constraint functions: we define the *Lagrangian* $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m_+ \times \mathbb{R}^l$ associated with the problem (A.1) as (Boyd and Vandenberghe, 2004, p. 215)

$$\mathcal{L}(\boldsymbol{v},\boldsymbol{\alpha},\boldsymbol{\beta}) = f_0(\boldsymbol{v}) + \sum_{i=1}^{m} \alpha_i f_i(\boldsymbol{v}) + \sum_{i=1}^{l} \beta_i h_i\,.$$

We refer to α_i as the Lagrangian multiplier associated with the ith inequality constraint and to β_i as the Lagrangian multiplier associated with the ith equality constraint. The *Lagrangian dual function* is defined as the minimum value of the Lagrangian over $\boldsymbol{v}$:

$$g(\boldsymbol{\alpha},\boldsymbol{\beta}) := \inf_{\boldsymbol{v}\in\mathbb{R}^n} L(\boldsymbol{v},\boldsymbol{\alpha},\beta)\,. \tag{A.2}$$

It is evident that the Lagrangian dual function yields a lower bound for the optimal value of the original problem (A.1); to see this, note that violated constraints contribute positively in (A.2). It is therefore recommended to maximize the lower bound. This raises the question "what is the *best* lower bound that can be obtained from the Lagrangian dual function?", leading to the optimization problem

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad & g(\boldsymbol{\alpha},\boldsymbol{\beta}) \\ \text{s.t.} \quad & \alpha_i \geq 0, \quad i = 1,\dots,m, \end{aligned}$$

which is known as the *dual problem* associated with (A.1). In this context the original problem is called *primal problem*. But how are the optima of the primal and dual problems related? For convex problems, we have the nice result that both optimal values are identical, under rather mild assumptions, which are known as *constraint qualifications* (CQs). For example, Slater's condition is such a CQ: there exists a point $\boldsymbol{v}$ such that $h_i(\boldsymbol{v}) = 0$ and $f_i(\boldsymbol{v}) < 0$ for all i active in the optimum $\boldsymbol{v}^*$ (Boyd and Vandenberghe, 2004, p. 225-226).

B Relating Tikhonov and Ivanov Regularization

In this section, 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 a key ingredient of the proof of Theorem 3.1. We state the result for arbitrary convex functions, so that it can be applied beyond the multiple kernel learning framework of this thesis.

Lemma B.1. *Let $D \subset \mathbb{R}^d$ be a convex set, let $f, g : D \to \mathbb{R}$ be convex functions. Consider the convex optimization tasks*

$$\min_{\boldsymbol{v} \in D} \quad f(\boldsymbol{v}) + \sigma g(\boldsymbol{v}), \tag{B.1a}$$

$$\min_{\boldsymbol{v} \in D : g(\boldsymbol{v}) \leq \tau} \quad f(\boldsymbol{v}). \tag{B.1b}$$

Assume that the minima exist and that a constraint qualification holds in (B.1b)*, which gives rise to strong duality, e.g., that Slater's condition is satisfied. Furthermore, assume that the constraint is active at the optimal point, i.e.*

$$\inf_{\boldsymbol{v} \in D} \; f(\boldsymbol{v}) \;<\; \inf_{\boldsymbol{v} \in D : g(\boldsymbol{v}) \leq \tau} \; f(\boldsymbol{v}). \tag{B.2}$$

Then we have the case that for each $\sigma > 0$ there exists $\tau > 0$—and vice versa—such that the optimization problem (B.1a) *is equivalent to* (B.1b)*, i.e., each optimal solution of one is an optimal solution of the other, and vice versa.*

Proof
(a). Let be $\sigma > 0$ and $\boldsymbol{v}^*$ be the optimal of (B.1a). We have to show that there exists a $\tau > 0$ such that $\boldsymbol{v}^*$ is optimal in (B.1b). We set $\tau = g(\boldsymbol{v}^*)$. Suppose $\boldsymbol{v}^*$ is not optimal in (B.1b), i.e., it exists $\tilde{\boldsymbol{v}} \in D : g(\tilde{\boldsymbol{v}}) \leq \tau$ such that $f(\tilde{\boldsymbol{v}}) < f(\boldsymbol{v}^*)$. Then we have

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

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

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

This contradicts the optimality of $\boldsymbol{v}^*$ in (B.1a), and hence proves that $\boldsymbol{v}^*$ is optimal in (B.1b), which was to be shown.

(b). Vice versa, let $\tau > 0$ be $\boldsymbol{v}^*$ optimal in (B.1b). The Lagrangian of (B.1b) is given by

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

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

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

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

$$\min_{\boldsymbol{v} \in D} \quad f(\boldsymbol{v}) + \sigma^*\left(g(\boldsymbol{v}) - \tau\right). \tag{B.3}$$

Removing the constant term $-\sigma^*\tau$, and setting $\sigma = \sigma^*$, we have that $\boldsymbol{v}^*$ is optimal in (B.1a), which was to be shown. Moreover by (B.2) we have that

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

and hence we see from Eq. (B.3) that $\sigma^* > 0$, which completes the proof of the proposition. ■

C Supplements to the Theoretical Analysis

The following result gives a block-structured version of Hölder's inequality (e.g., Steele, 2004).

Lemma C.1 (Block-structured Hölder inequality)**.** *Let* $\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_M)$, $\boldsymbol{w} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M) \in \mathcal{H} = \mathcal{H}_1 \times \cdots \times \mathcal{H}_M$*. Then, for any* $p \geq 1$*, it holds*

$$\langle \boldsymbol{v}, \boldsymbol{w} \rangle \leq \|\boldsymbol{v}\|_{2,p} \|\boldsymbol{w}\|_{2,p^*} \ .$$

Proof By the Cauchy-Schwarz inequality (C.-S.), we have for all $x, \boldsymbol{y} \in \mathcal{H}$:

$$\begin{aligned} \langle \boldsymbol{v}, \boldsymbol{w} \rangle \quad &= \quad \sum_{m=1}^{M} \langle \boldsymbol{v}_m, \boldsymbol{w}_m \rangle \overset{\text{C.-S.}}{\leq} \sum_{m=1}^{M} \|\boldsymbol{v}_m\|_2 \|\boldsymbol{w}_m\|_2 \\ &= \quad \big\langle (\|\boldsymbol{v}_1\|_2, \ldots, \|\boldsymbol{v}_M\|_2), (\|\boldsymbol{w}_1\|_2, \ldots, \|\boldsymbol{w}_M\|_2) \big\rangle. \\ &\overset{\text{Hölder}}{\leq} \quad \|\boldsymbol{v}\|_{2,p} \|\boldsymbol{w}\|_{2,p^*} \end{aligned}$$

■

Proof of Lemma 4.3 (Rosenthal + Young) It is clear that the result trivially holds for $\frac{1}{2} \leq p \leq 1$ with $c_q = 1$ by Jensen's inequality . In the case $p \geq 1$, we apply

Rosenthal's inequality (Rosenthal, 1970) to the sequence $X_1, \ldots, X_n$ thereby using the optimal constants computed in Ibragimov and Sharakhmetov (2001), that are, $c_q = 2$ ($q \leq 2$) and $c_q = \mathbb{E}Z^q$ ($q \geq 2$), respectively, where Z is a random variable distributed according to a Poisson law with parameter $\lambda = 1$. This yields

$$\mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right)^q \leq c_q \max\left(\frac{1}{n^q}\sum_{i=1}^{n}\mathbb{E}X_i^q, \left(\frac{1}{n}\sum_{i=1}^{n} X_i\right)^q\right). \tag{C.1}$$

By using the fact that $X_i \leq B$ holds almost surely, we could readily obtain a bound of the form $\frac{B^q}{n^{q-1}}$ on the first term. However, this is loose and for $q = 1$ does not converge to zero when $n \to \infty$. Therefore, we follow a different approach based on Young's inequality (e.g. Steele, 2004):

$$\begin{aligned}
\frac{1}{n^q}\sum_{i=1}^{n}\mathbb{E}X_i^q &\leq \left(\frac{B}{n}\right)^{q-1}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}X_i \\
&\overset{\text{Young}}{\leq} \frac{1}{q^*}\left(\frac{B}{n}\right)^{q^*(q-1)} + \frac{1}{q}\left(\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}X_i\right)^q \\
&= \frac{1}{q^*}\left(\frac{B}{n}\right)^{q} + \frac{1}{q}\left(\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}X_i\right)^q.
\end{aligned}$$

It thus follows from (C.1) that for all $q \geq \frac{1}{2}$

$$\mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right)^q \leq c_q\left(\left(\frac{B}{n}\right)^q + \left(\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}X_i\right)^q\right),$$

where c_q can be taken as 2 ($q \leq 2$) and $\mathbb{E}Z^q$ ($q \geq 2$), respectively, where Z is Poisson-distributed. In the subsequent Lemma C.2 we show $\mathbb{E}Z^q \leq (q+e)^q$. Clearly, for $q \geq \frac{1}{2}$ it holds $q + e \leq qe + eq = 2eq$ so that in any case $c_q \leq \max(2, 2eq) \leq 2eq$, which completes the proof. ■

We use the following Lemma, which gives a handle on the q-th moment of a Poisson-distributed random variable and is used in the previous Lemma.

Lemma C.2. *For the q-moment of a random variable Z distributed according to a Poisson law with parameter $\lambda = 1$, the following inequality holds for all $q \geq 1$:*

$$\mathbb{E}Z^q \overset{def.}{=} \frac{1}{e}\sum_{k=0}^{\infty}\frac{k^q}{k!} \leq (q+e)^q.$$

Proof We start by decomposing $\mathbb{E}Z^q$ as follows:

$$\begin{aligned}\mathbb{E}^q &= \frac{1}{e}\left(0+\sum_{k=1}^{q}\frac{k^q}{k!}+\sum_{k=q+1}^{\infty}\frac{k^q}{k!}\right)\\ &= \frac{1}{e}\left(\sum_{k=1}^{q}\frac{k^{q-1}}{(k-1)!}+\sum_{k=q+1}^{\infty}\frac{k^q}{k!}\right)\\ &\leq \frac{1}{e}\left(q^q+\sum_{k=q+1}^{\infty}\frac{k^q}{k!}\right) \qquad \text{(C.2)}\\ & \qquad \text{(C.3)}\end{aligned}$$

Note that by Stirling's approximation it holds $k! = \sqrt{2\pi}e^{\tau_k}k\left(\frac{k}{e}\right)^q$ with $\frac{1}{12k+1} < \tau_k < \frac{1}{12k}$ for all q. Thus

$$\begin{aligned}\sum_{k=q+1}^{\infty}\frac{k^q}{k!} &= \sum_{k=q+1}^{\infty}\frac{1}{\sqrt{2\pi}e^{\tau_k}k}e^k k^{-(k-q)}\\ &= \sum_{k=1}^{\infty}\frac{1}{\sqrt{2\pi}e^{\tau_{k+q}}(k+q)}e^{k+q}k^{-k}\\ &= e^q\sum_{k=1}^{\infty}\frac{1}{\sqrt{2\pi}e^{\tau_{k+q}}(k+q)}\left(\frac{e}{k}\right)^k\\ &\overset{(*)}{\leq} e^q\sum_{k=1}^{\infty}\frac{1}{\sqrt{2\pi}e^{\tau_k}k}\left(\frac{e}{k}\right)^k\\ &\overset{\text{Stirling}}{=} e^q\sum_{k=1}^{\infty}\frac{1}{k!}\\ &= e^{q+1}\end{aligned}$$

where for $(*)$ note that $e^{\tau_k}k \leq e^{\tau_{k+q}}(k+q)$ can be shown by some algebra using $\frac{1}{12k+1} < \tau_k < \frac{1}{12k}$. Now by (C.2)

$$\mathbb{E}Z^q = \frac{1}{e}\left(q^q+e^{q+1}\right) \leq q^q+e^q \leq (q+e)^q,$$

which was to show. ∎

Lemma C.3. *For any $\boldsymbol{a},\boldsymbol{b}\in\mathbb{R}^m_+$ it holds for all $q\geq 1$*

$$\|\boldsymbol{a}\|_q + \|\boldsymbol{b}\|_q \leq 2^{1-\frac{1}{q}}\|\boldsymbol{a}+\boldsymbol{b}\|_q \leq 2\|\boldsymbol{a}+\boldsymbol{b}\|_q.$$

Proof Let $\boldsymbol{a} = (a_1, \ldots, a_m)$ and $\boldsymbol{b} = (b_1, \ldots, b_m)$. Because all components of $\boldsymbol{a}, \boldsymbol{b}$ are nonnegative, we have

$$\forall i = 1, \ldots, m: \; a_i^q + b_i^q \leq \big(a_i + b_i\big)^q$$

and thus

$$\|\boldsymbol{a}\|_q^q + \|\boldsymbol{b}\|_q^q \leq \|\boldsymbol{a} + \boldsymbol{b}\|_q^q \,. \tag{C.4}$$

We conclude by ℓ_q-to-ℓ_1 conversion (see (4.15))

$$\begin{aligned} \|\boldsymbol{a}\|_q + \|\boldsymbol{b}\|_q &= \big\|\big(\,\|\boldsymbol{a}\|_q, \|\boldsymbol{b}\|_q\,\big)\big\|_1 \overset{(4.15)}{\leq} 2^{1-\frac{1}{q}} \big\|\big(\,\|\boldsymbol{a}\|_q, \|\boldsymbol{b}\|_q\,\big)\big\|_q \\ &= 2^{1-\frac{1}{q}} \big(\,\|\boldsymbol{a}\|_q^q + \|\boldsymbol{b}\|_q^q\,\big)^{\frac{1}{q}} \overset{(C.4)}{\leq} 2^{1-\frac{1}{q}} \,\|\boldsymbol{a} + \boldsymbol{b}\|_q \,, \end{aligned}$$

which completes the proof. ■

D Cutting Plane Algorithm

In this section, we present an alternative optimization strategy for the case $p \in [1, 2]$, based on the cutting plane method; to this end, we denote $q = p/(2-p)$ (note that thus $q \in [1, \infty]$ since $p \in [1, 2]$). Our algorithm is based on the MKL optimization problem II.4. We focus on the hinge loss, i.e., $l(t, y) = \max(0, 1 - ty)$. Thus, employing a variable substitution of the form $\alpha_i^{\text{new}} = \alpha_i y_i$, Problem II.4 translates into

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

where $Q_j = Y K_j Y$ for $1 \leq j \leq m$ and $Y = \operatorname{diag}(\mathbf{y})$. The above optimization problem is a *saddle point problem* and, as we have seen, can be solved by a block coordinate descent algorithm, using an analytical formula. 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 program:

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

Algorithm D.1 (CHUNKING CPM) 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.

$g_{m,i} = 0$, $\hat{g}_i = 0$, $\alpha_i = 0$, $\eta = -\infty$, $\theta_m = \sqrt[q]{1/M}$ 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 $g_{m,i} \leftarrow g_{m,i} + \sum_{l=1}^{Q} (\alpha_{i_l} - \alpha_{i_l}^{old}) y_{i_l} k_m(x_{i_l}, x_i)$ 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 $|1 - \frac{L^t - S^t}{\eta}| \geq \epsilon$
    while MKL optimality conditions are not satisfied **do**
      $\boldsymbol{\theta}^{old} = \boldsymbol{\theta}$
      $(\boldsymbol{\theta}, \eta) \leftarrow$ argmin η
        w.r.t. $\boldsymbol{\theta} \in \mathbb{R}^M, \eta \in \mathbb{R}$
        s.t. $\mathbf{0} \leq \boldsymbol{\theta} \leq \mathbf{1}$,
          $\frac{q(q-1)}{2} \sum_m (\theta_m^{old})^{q-2} \theta_m^2 - \sum_m q(q-2)(\theta_m^{old})^{q-1} \theta_m \leq \frac{q(3-q)}{2}$ and
          $L^r - \sum_m \theta_m S_m^r \leq \eta$ for $r = 1, \ldots, t$
      $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} / ||\boldsymbol{\theta}||_q^q$
      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

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 $q > 1$, a non-linearity is introduced by requiring $\|\boldsymbol{\theta}\|_q^q \leq 1$ 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 $\|\boldsymbol{\theta}\|_q^q \leq 1$ by a sequence of second-order Taylor expansions[9]

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

where $\boldsymbol{\theta}^q$ is defined element-wise, that is $\boldsymbol{\theta}^q := (\theta_1^q, \ldots, \theta_M^q)$. The sequence $(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \cdots)$ is initialized with a uniform mixture satisfying $||\boldsymbol{\theta}_0||_q^q = 1$ 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 positive and very-well conditioned, for which reason 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 enhance the resulting sequential quadratically constrained quadratic programming by projection steps onto the boundary of the feasible set, as given in Line 18. 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 D.1 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 $|1 - \frac{L-S^t}{\eta}| < \varepsilon_{mkl}$ for the MKL. The following proposition shows the convergence of the semi-infinite programming loop in Algorithm D.1.

Proposition D.1. *Let the kernel matrices $K_1, \ldots, K_M$ be positive-definite and be $q > 1$. 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 D.1 has at least one point of accumulation that solves* (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 D.1 can be ignored. We conclude that the outer loop of Alg. D.1 amounts to a cutting-plane algorithm for solving the semi-infinite program of Optimization Problem (SIP). It is well

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

known (Sonnenburg et al., 2006a) that this algorithm converges, in the sense that there exists at least one point of accumulation that 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 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_q-norm training algorithm, for all $q \in (1, \infty)$ (Fernández and Solodov, 2008) and $q \geq 2$ (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 with 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. D.1 would guarantee the convergence of our Alg. D.1.

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. D.1, 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, to the best of our knowledge, projection-type algorithms for solving SQCQPs have not been studied yet by the optimization literature.

Although the projection procedure is appealingly simple and—as we found out 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 discus-

sions 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 converge. The theoretical analysis of this new class of SQCQP projection algorithms is beyond the scope of this thesis.

Bibliography

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*, 12:565–592, Feb 2011.

B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular biology of the cell*. Garland Science, 4 edition, 2002. ISBN 0815332181.

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

F. Bach. Eccv 2008 tutorial on supervised learning, 2008a.

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

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

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

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

P. L. Bartlett and S. Mendelson. Empirical minimization. *Probability Theory and Related Fields*, 135(3):311–334, 2006.

P. L. Bartlett, O. Bousquet, and S. Mendelson. Local Rademacher complexities. *Annals of Statistics*, 33(4):1497–1537, 2005.

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

A. Binder, K.-R. Müller, and M. Kawanabe. On taxonomies for multi-class image categorization. *International Journal of Computer Vision*, January 2011.

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

A. Bosch, A. Zisserman, and X. Muñoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval (CIVR '07)*, pages 401–408, 2007. ISBN 978-1-59593-733-9.

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

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

C. Campbell and Y. Ying. *Learning with Support Vector Machines*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2011.

J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8:679–714, 1986.

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, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006. URL `http://www.kyb.tuebingen.mpg.de/ssl-book`.

C. Cortes. Invited talk: Can learning kernels help performance? In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1:1–1:1, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. Video `http://videolectures.net/icml09_cortes_clkh/`.

C. Cortes, A. Gretton, G. Lanckriet, M. Mohri, A. Rostamizadeh, and editors, editors. *Proceedings of the NIPS 2008 workshop on kernel learning: automatic selection of optimal kernels*, 2008. URL `http://www.cs.nyu.edu/learning_kernels`, Video `http://videolectures.net/lkasok08_whistler/`.

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

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

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

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

N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In T. Fawcett and N. Mishra, editors, *Advances in Neural Information Processing Systems*, volume 14, pages 367–373, 2002.

G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, volume 2, pages 1–22, Prague, Czech Republic, May 2004.

N. Dalal and B. Triggs. Histograms of oriented gradientsfor human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, San Diego, USA, June 2005.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results, 2008. URL `http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html`.

T. Evgeniou and M. Pontil. Regularized multi–task learning. In W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel, editors, *KDD*, pages 109–117. ACM, 2004. ISBN 1-58113-888-1.

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.

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.

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.

P. V. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, pages 221–228, 2009.

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

N. Görnitz, M. Kloft, and U. Brefeld. Active and semi-supervised data domain description. 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 407–422, 2009a.

N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld. Active learning for network intrusion detection. In *Proc. of CCS Workshop on Security and Artificial Intelligence (AISEC)*, pages 47–54, New York, NY, USA, 2009b. ACM.

N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld. Active and semi-supervised domain descriptions in theory and practice. *Journal of Artificial Intelligence Research (JAIR)*, 2010. submitted 07/2010.

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

R. Ibragimov and S. Sharakhmetov. The best constant in the rosenthal inequality for non-negative random variables. *Statistics & Probability Letters*, 55(4):367 – 376, 2001. ISSN 0167-7152.

R. Jenssen, M. Kloft, A. Zien, S. Sonnenburg, and K.-R. Müller. A multi-class support vector machine based on scatter criteria. Technical Report 014-2009, Technische Universität Berlin, Jul 2009. Revised version submitted to Pattern Recognition, Apr 2011.

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.

J.-P. Kahane. *Some random series of functions*. Cambridge University Press, 2nd edition, 1985.

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 and G. Blanchard. The local rademacher complexity of lp-norm multiple kernel learning. Technical report, Arxiv preprint 1103.0790v1, Mar 2011. URL `http://arxiv.org/abs/1103.0790v1`. Full version submitted to *Journal of Machine Learning Research*, Mar 2011. Short Version submitted to *Twenty-Fifth Annual Conference on Neural Information Processing Systems (NIPS 2011)*.

M. Kloft and P. Laskov. Online anomaly detection under adversarial impact. In Y. W. Teh and M. Titterington, editors, *JMLR Workshop and Conference Proceedings, Volume 9: AISTATS*, pages 405–412. MIT Press, 2010a.

M. Kloft and P. Laskov. Security analysis of online centroid anomaly detection. Technical Report UCB/EECS-2010-22, EECS Department, University of California, Berkeley, Feb 2010b. URL `http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-22.html`. CoRR abs/1003.0078. Revised version submitted to Journal of Machine Learning Resarch, Feb 2011.

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, 2008a.

M. Kloft, U. Brefeld, P. Laskov, and S. Sonnenburg. Non-sparse multiple kernel learning. In *Proc. of the NIPS Workshop on Kernel Learning: Automatic Selection of Kernels*, Dec 2008b. URL `http://www.cs.nyu.edu/learning_kernels/abstracts/ws_mkl.pdf`.

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

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

M. Kloft, U. Rückert, and P. L. Bartlett. A unifying view of multiple kernel learning. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pages 66–81, 2010a.

M. Kloft, U. Rückert, C. S. Ong, A. Rakotomamonjy, S. Sonnenburg, and F. Bach, editors. *Proceedings of the NIPS 2010 workshop on new directions in multiple kernel learning*, 2010b. URL `http://doc.ml.tu-berlin.de/mkl_workshop/`, Video `http://videolectures.net/nipsworkshops2010_kernel_learning/`.

M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. Lp-norm multiple kernel learning. *Journal of Machine Learning Research*, 12:953–997, Mar 2011.

V. Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001.

V. Koltchinskii. Local Rademacher complexities and oracle inequalities in risk minimization. *Annals of Statistics*, 34(6):2593–2656, 2006.

V. Koltchinskii. Sparsity in penalized empirical risk minimization. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 45(1):7–57, 2009.

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

V. Koltchinskii and M. Yuan. Sparsity in multiple kernel learning. *Annals of Statistics*, 38(6): 3660–3695, 2010.

S. Kwapién and W. A. Woyczyński. *Random Series and Stochastic Integrals: Single and Multiple*. Birkhäuser, Basel and Boston, M.A., 1992.

C. Lampert and M. Blaschko. A multiple kernel learning approach to joint multi-class object detection. In *DAGM*, pages 31–40, 2008.

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, 2004a.

G. Lanckriet, F. Bach, N. Srebro, and B. McFee, editors. *Proceedings of the NIPS 2009 Workshop on Understanding Multiple Kernel Learning Methods*, 2009. URL `http://mkl.ucsd.edu/workshop`.

G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20:2626–2635, November 2004b. ISSN 1367-4803.

P. Laskov and M. Kloft. A framework for quantitative security analysis of machine learning. In D. Balfanz and J. Staddon, editors, *Proceedings of the 2nd ACM Workshop on Security and Artificial Intelligence (AISEC)*, pages 1–4, 2009.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, volume 2, pages 2169–2178, New York, USA, 2006.

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

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

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.

M. Marszalek and C. Schmid. Learning representations for visual object class recognition. In *Proceedings of the PASCAL Visual Object Classes Challenge 2007 (VOC2007)*, 2007.

A. F. Martins, M. A. Figueiredo, P. M. Aguiar, N. A. Smith, and E. P. Xing. Online learning of structured predictors with multiple kernels. In *JMLR Workshop and Conference Proceedings: AISTATS 2011*. MIT Press, 2011. To appear.

S. Mendelson. On the performance of kernel classes. *J. Mach. Learn. Res.*, 4:759–771, December 2003.

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*, 2009a.

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*, 2009b.

S. Nakajima, A. Binder, C. Müller, W. Wojcikiewicz, M. Kloft, U. Brefeld, K.-R. Müller, and M. Kawanabe. Multiple kernel learning for object classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010. submitted 09/2010.

W. S. Noble. Multi-kernel learning for biology. In *Proceedings of the NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels*, 2008. URL `http://videolectures.net/lkasok08_noble_mklfb/`.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

G. Obozinski, M. J. Wainwright, and M. I. Jordan. Support union recovery in high-dimensional multivariate regression. *Ann. Stat.*, 39(1):1–47, 2011.

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.

F. Orabona and L. Jie. Ultra-fast optimization algorithm for sparse multi kernel learning. In *Proceedings of the 28th Annual International Conference on Machine Learning*, ICML '11, 2011. To appear.

F. Orabona, J. Luo, and B. Caputo. Online-batch strongly convex multi kernel learning. In *CVPR*, pages 787–794. IEEE, 2010.

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

G. Raskutti, M. J. Wainwright, and B. Yu. Minimax-optimal rates for sparse additive models over kernel classes via convex programming. *CoRR*, abs/1008.3654, 2010.

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

H. Rosenthal. On the subspaces of L_p $(p > 2)$ spanned by sequences of independent random variables. *Israel J. Math.*, 8:273–303, 1970.

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.

U. Rückert and M. Kloft. Transfer learning with adaptive regularizers. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, 2011.

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.

B. Schölkopf, K. Tsuda, and J. P. Vert, editors. *Kernel Methods in Computational Biology*. MIT Press, 2004.

J. R. Searle. Minds, brains, and programs. *Behavioral and Brain Sciences*, 3(03):417–424, 1980.

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

M. Sion. On general minimax theorems. *Pacific J. Math.*, 8(1):171–176, 1958.

M. V. Solodov. On the sequential quadratically constrained quadratic programming methods. *Math. Oper. Res.*, 29(1):64–79, 2004.

S. Sonnenburg. *Machine Learning for Genomic Sequence Analysis*. PhD thesis, Fraunhofer Institute FIRST, December 2008.

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

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

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

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

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. *Mach. Learn.*, 79(1-2):73–103, 2010.

A. Tsybakov. Optimal rates of aggregation. In B. Schölkopf and M. Warmuth, editors, *Computational Learning Theory and Kernel Machines (COLT-2003)*, volume 2777 of *Lecture Notes in Artificial Intelligence*, pages 303–313. Springer, 2003.

V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, Sept. 1998.

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.

U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

C. Widmer, N. Toussaint, Y. Altun, and G. Ratsch. Inferring latent task structure for multitask learning by multiple kernel learning. *BMC Bioinformatics*, 11(Suppl 8):S5, 2010.

R. C. Williamson, A. J. Smola, and B. Schölkopf. Generalization performance of regularization networks and support vector machines via entropy numbers of compact operators. *IEEE Transactions on Information Theory*, 47(6):2516–2532, 2001.

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, M. Barnard, H. Cai, and J. Kittler. Lp norm multiple kernel fisher discriminant analysis for object and image categorisation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2010.

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

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

J. Zhang, M. Marszalek, S.Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007.

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.

Curriculum Vitae

Personal Information

Name	Marius Kloft
Date of birth	July 16, 1980
Place of birth	Dernbach, Rhineland-Palatinate, Germany
Citizenship	German
Email	`mkloft@mail.tu-berlin.de`

Education and Employment

since 01/2007	PhD studies in Computer Science at the Machine Learning Laboratory, Department of EECS, *Berlin Institute of Technology (TU Berlin)*. Advisor: Prof. Dr. Klaus-Robert Müller.
since 10/2010	PhD scholarship by the *Berlin Institute of Technology (TU Berlin)*.
10/2009–09/2010	Research stay at the Learning Theory Group, EECS department, *University of California*, Berkeley, USA. Founded by a scholarship of the German Foreign Exchange Service (DAAD). Advisor: Prof. Dr. Peter L. Bartlett.
06/2007–09/2009	Employment in the BMBF project *Real-time Machine Learning for Intrusion Detection* (REMIND) in collaboration with the *Fraunhofer FIRST Institute*, Berlin. Supervisor: Pavel Laskov, PhD.
01/2007–04/2007	Employment in the BMBF project *FASOR*.
07/2006	Diploma in mathematics (M.Sc. equivalent) with GPA 1.0 (1.0 is max). Diploma thesis in algebraic geometry on *Stable Base Loci and the Theorem of Zariski-Fujita*. Advisor: Prof. Dr. Thomas Bauer.
10/2001–07/2006	Studies in mathematic with minor computer science at the *University of Marburg*, Germany. Majors: Algebraic geometry, algebra, complex geometry, probability theory, universal algebra.
10/2000–09/2001	Studies in physics at the *University of Marburg*, Germany.
09/2000	Internship at Klöckner Pentaplast of America, Gordonsville, VA, USA.
07/2000	Award by the German physical society for achievements in the high school major physics.
08/1989–07/2000	Mons-Tabor Gymnasium Montabaur (high school)

Publications

Publications related to this thesis are marked with •, others with ◦.

Preprints (submitted)

◦ D. Deb, J. Lässig, and **M. Kloft**. Effects of Seedling Age and Land Type on Grain Yield in the System of Rice Intensification. Submitted to *Experimental Agriculture*, Jun 2011. [27]

• **M. Kloft** and G. Blanchard. The Local Rademacher Complexity of ℓ_p-Norm Multiple Kernel Learning. ArXiv preprint 1103.0790, Mar 2011. Submitted to *Journal of Machine Learning Research (JMLR)*. Status: *Accepted pending minor revision*. [26]

◦ R. Jenssen, **M. Kloft**, A. Zien, S. Sonnenburg, and K.-R. Müller. A Multi-Class Support Vector Machine based on Scatter Criteria. Technical Report 14/2009, Technische Universität Berlin, Jun 2009. Submitted to *Pattern Recognition*. Status: *revision*. Revision submitted, Apr 2011. [25]

◦ **M. Kloft** and P. Laskov. Security Analysis of Online Centroid Anomaly Detection. Technical Report UCB/EECS-2010-22, University of California, Berkeley, Feb 2010. Submitted to *Journal of Machine Learning Research (JMLR)*. Status: *Accepted pending minor revision*. [24]

• A. Binder, S. Nakajima, **M. Kloft**, C. Müller, W. Wojcikiewicz, U. Brefeld, K.-R. Müller, and M. Kawanabe. Classifying Visual Objects with many Kernels. Submitted to *Transactions of Pattern Analysis and Machine Intelligence (TPAMI)*. Status: *major revision*. Revision in preparation. [23]

◦ N. Görnitz, **M. Kloft**, K. Rieck, and U. Brefeld. From Unsupervised towards Supervised Anomaly Detection. Submitted to *Journal of Artificial Intelligence Research (JAIR)*. Status: *major revision*. Revision in preparation. [22]

◦ S. Sonnenburg, C. S. Ong, F. Bach, **M. Kloft**, U. Rückert, and A. Rakotomamonjy. Introduction to the JMLR Special Topic on Metric and Kernel Learning. *Journal of Machine Learning Research (JMLR)*. Editorial paper. Manuscript in preparation. [21]

2011

• **M. Kloft** and G. Blanchard. The Local Rademacher Complexity of ℓ_p-Norm Multiple Kernel Learning. *Advances in Neural Information Processing Systems 24 (NIPS)*, to appear. [20]

◦ A. Binder, S. Samek, **M. Kloft**, C. Müller, K.-R. Müller, and M. Kawanabe. The joint submission of the TU Berlin and Fraunhofer FIRST (TUBFI) to the ImageCLEF2011 Photo Annotation Task. *Proceedings of the 12th Workshop of the Cross-Language Evaluation Forum (CLEF 2011)*, (to appear) 2011. [19]

◦ R. Jenssen, **M. Kloft**, A. Zien, S. Sonnenburg, and K.-R. Müller. A New Scatter-Based Multi-Class Support Vector Machine. *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, (to appear) 2011. [18]

◦ U. Rückert and **M. Kloft**. Transfer Learning with Adaptive Regularizers. *Proceedings of the European Conference on Machine Learning (ECML)*, (to appear) 2011. [17]

• **M. Kloft**, U. Brefeld, S. Sonnenburg, and A. Zien. ℓ_p-Norm Multiple Kernel Learning. *Journal of Machine Learning Research (JMLR)*, volume 12, pages 953-997, 2011. [16]

2010

◦ **M. Kloft**, U. Rückert, C. S. Ong, A. Rakotomamonjy, S. Sonnenburg, and F. Bach, editors, *Proceedings of the NIPS 2010 Workshop on New Directions in Multiple Kernel Learning*, 2010. [15]

• **M. Kloft**, U. Rückert, and P. L. Bartlett. A Unifying View of Multiple Kernel Learning. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 66–81, 2010. [14]

◦ **M. Kloft** and P. Laskov. Online Anomaly Detection under Adversarial Impact. In *JMLR Workshop and Conference Proceedings 9 (AISTATS 2010)*, pages 405-412. MIT Press, 2010. [13]

2009

• **M. Kloft**, U. Brefeld, S. Sonnenburg, and A. Zien. Comparing Sparse and Non-sparse Multiple Kernel Learning. *Proceedings of the NIPS 2009 Workshop on Understanding Multiple Kernel Learning Methods*, 2009. [12]

• A. Binder, M. Kawanabe, **M. Kloft**, and S. Nakajima. Enhancing Image Annotation with Primitive Color Histograms via Non-sparse Multiple Kernel Learning. In *Proceedings of the NIPS 2009 Workshop on Understanding Multiple Kernel Learning Methods*, 2009. [11]

• **M. Kloft**, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Müller, and A. Zien. Efficient and Accurate ℓ_p-norm Multiple Kernel Learning. In *Advances in Neural Information Processing Systems 22 (NIPS)*, pages 997-1005, 2009.[10] [10]

[10] 54 citations, `http://scholar.google.com`, June 19, 2011.

• 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 (IBIS)*, 2009. [9]

◦ N. Görnitz, **M. Kloft**, K. Rieck, and U. Brefeld. Active Learning for Network Intrusion Detection. In *Proceedings of the 2nd ACM Workshop on Security and Artificial Intelligence (AISec)*, pages 47-54. ACM, 2009. [8]

◦ P. Laskov and **M. Kloft**. A Framework for Quantitative Security Analysis of Machine Learning. In *Proceedings of the 2nd ACM Workshop on Security and Artificial Intelligence (AISec)*, pages 1-4. ACM, 2009. [7]

◦ N. Görnitz, **M. Kloft**, and U. Brefeld. Active and Semi-supervised Data Domain Description. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pages 407-422, 2009. [6]

• **M. Kloft**, S. Nakajima, and U. Brefeld. Feature selection for density level-sets. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pages 692-704, 2009. [5]

• **M. Kloft**, U. Brefeld, S. Sonnenburg, A. Zien, P. Laskov, and K.-R. Müller. Learning Non-sparse Kernel Mixtures. In *Proceedings of the PASCAL2 Workshop on Sparsity in Machine Learning and Statistics*, 2009. [4]

2008

• **M. Kloft**, U. Brefeld, P. Laskov, and S. Sonnenburg. Non-sparse Multiple Kernel Learning. In *Proceedings of the NIPS 2008 Workshop on Kernel Learning: Automatic Selection of Optimal Kernels*, 2008. [3]

• **M. Kloft**, U. Brefeld, P. Düssel, C. Gehl, and P. Laskov. Automatic feature selection for anomaly detection. In *Proceedings of the 1st ACM Workshop on Security and Artificial Intelligence (AISec)*, pages 71-76. ACM, 2008. [2]

◦ **M. Kloft** and P. Laskov. A Poisoning Attack Against Online Anomaly Detection. In *Proceedings of the NIPS 2007 Workshop on Machine Learning in Adversarial Environments*, 2008. [1]