Theoretical Foundations and Algorithms for Learning with Multiple Kernels

by

Afshin Rostamizadeh

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

Courant Institute of Mathematical Sciences

New York University

May 2010

Mehryar Mohri—Advisor

To my family.

Acknowledgments

It would not have been possible for me to accomplish this difficult and exciting journey without the help of many people who have been important both in my professional and personal life.

Without the careful guidance and knowledge of my advisor, Mehryar Mohri, the task of completing my PhD studies would have been an infinitely more difficult one. His insight into interesting problems and how to effectively attack them has not only helped me in writing this thesis, but has also taught me how to really conduct research. I am fortunate to have worked on several diverse research topics with him. I thank him for his patience and the great effort he has made in providing me with opportunities that I am lucky to have.

During the course of my studies, I have been able to complete several summer internships at Google Research. These internships have been very influential in my graduate career and have given me the opportunity to work with several great hosts: Michael Riley, Corinna Cortes and Cyril Allauzen. I have had the privilege to co-author papers with all of these world-class researchers and have learned many useful lessons as a result. I would especially like to thank Corinna Cortes, who is also a reader for this thesis, and who has worked very closely on the topic of learning with multiple kernels. Her ability to find and correct flaws in empirical evaluations and intuition for improving algorithms have helped me tremendously.

I would also like to thank another co-author and defense committee member, Yishay Mansour, for allowing me to work with him on several interesting problems. His ability to provide simple intuition for complex problems, as well as his constant positivity and excitement, are greatly appreciated. I also thank Subhash Khot and Joel Spencer for sitting on my defense committee as well as my thesis proposal and depth qualification exam; your time and effort are appreciated.

My experience in graduate school has been made much more enjoyable in the company of other great students. Ameet Talwalkar was a fellow student I met as soon as I arrived at NYU and little did I know that we would share the same classes, office and advisor for the coming years. I am very happy to have had such an encouraging and helpful confidant, who has been available to discuss both research and personal matters. I want to especially thank Deep Ganguli and Dejan Jovanovic, my closest friends in these last few years. Their sense of humor, their excitement for exploring New York City and their loyal friendship will never be forgotten. I am fortunate to share a department with so many more friends, Shaila Musharoff, Mina Jeong and Chris Conway, to name a few, and I hope to keep in contact for many years to come.

Finally, nobody has had as much impact in my life as my loving mother and father, Nasrin and Abbas, who have always helped me achieve my greatest potential. They have always believed in me and I have enjoyed their unwaivering support at all stages of my academic and personal life. My brother Aria is one of my favorite people and a close friend. I think we share a sense of humor that may be difficult for anyone else to understand, and I thank him for always bringing a smile to my face. I hope to have made them all proud of this accomplishment and will strive to continue doing so in the future.

Abstract

Kernel-based algorithms have been used with great success in a variety of machine learning applications. These include algorithms such as support vector machines for classification, kernel ridge regression, ranking algorithms, clustering algorithms, and virtually all popular dimensionality reduction algorithms.

But, the choice of the kernel, which is crucial to the success of these algorithms, has been traditionally left entirely to the user. Rather than requesting the user to commit to a specific kernel, multiple kernel algorithms require the user only to specify a family of kernels. This family of kernels can be used by a learning algorithm to form a combined kernel and derive an accurate predictor. This is a problem that has attracted a lot of attention recently, both from the theoretical point of view and from the algorithmic, optimization, and application point of view.

This thesis presents a number of novel theoretical and algorithmic results for learning with multiple kernels.

It gives the first tight margin-based generalization bounds for learning kernels with L_p regularization. In particular, our margin bounds for L_1 regularization are shown to have only a logarithmic dependency on the number of kernels, which is a significant improvement over all previous analyses. Our results also include stability-based guarantees for a class of regression algorithms. In all cases, these guarantees indicate the benefits of learning with a large number of kernels.

We also present a family of new two-stage algorithms for learning kernels based on a notion of alignment and give an extensive analysis of the properties of these algorithms. We show the existence of good predictors for the notion of alignment we define and give efficient algorithms for learning a maximum alignment kernel by showing that the problem can be reduced to a simple quadratic program.

Finally, we report the results of extensive experiments with our two-stage algorithms, which show an improvement both over the uniform combination of kernels and over other state-of-the-art learning kernel methods for L_1 and L_2 regularization. These might constitute the first series of results for learning with multiple kernels that demonstrate a consistent improvement over a uniform combination of kernels.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In machine learning the goal is to learn from labeled examples in order to predict the labels of other, possibly never before seen, unlabeled examples. That is, given a training set, we wish to learn a rule, or hypothesis, that will *generalize* well. In order to do this, each instance is first represented by a set of features, which often represents one's prior knowledge about the task. The learning algorithm will make use of these features to select a hypothesis. Correctly selecting this hypothesis to minimize errors on previously unseen points, is then the main focus of machine learning algorithms. These are the two steps illustrated in Diagram (1.1).

As is shown in the diagram, the choice of features for an instance $x \in \mathcal{X}$, denoted by a vector $\Phi(x)$, is often made before the traditional "learning" step where a hypothesis h is chosen based on labeled training examples. Thus, the choice of features will have a direct impact on the selection and performance of the hypothesis h.

$$\begin{array}{ccccc} x & \rightarrow & \Phi(x) & \rightarrow & h(\Phi(x)) \\ \text{instance} & & \text{features} & & \text{prediction} \end{array} \tag{1.1}$$

To give a simple albeit extreme example that illustrates the importance of features, consider two cases: one in which the features in fact contain the correct prediction label and another in which the features contain random values. In such scenarios the choice of learning algorithm is rather moot. In the former case any reasonable algorithm will do well, while in the latter no algorithm can be expected to perform better than random guessing.

Traditionally, it is the user's task to define a set of useful features. This requires the user to have some prior knowledge about which aspects of the data will be useful for predicting labels. For example, if the task is to distinguish between apples and oranges, a very useful feature would be the color of the fruit. In many modern machine learning algorithms features can be represented either explicitly, such as $\Phi(x)$, or implicitly via a kernel function $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$. Kernel functions define a similarity measure between instances and will allow for flexibility and efficiency in representing features. They will be discussed more fully in Section 1.3. In real world scenarios, the task of correctly choosing a set of features or a kernel is often non-trivial and committing to a set of useful features may be a very difficult task.

Can we instead provide algorithms and guarantees that will help us find useful features? Can we leverage the flexibility and efficiency of kernel functions, with their ability to define diverse and powerful feature spaces, in order to effectively search and find a “good” set of features? Can the burden of defining good features be lessened for the user? In this thesis, we will consider algorithms which not only select the hypothesis, but also the kernel from a given family of kernels. Thus, the requirement on the user is reduced to selecting a general family of kernels instead of committing to a single kernel.

Such methods, often referred to as “learning kernel” methods, have been previously proposed and are discussed fully in Section 1.4, however, the problem is far from solved. In practice, these automated methods are not always able to improve upon simple baselines or heuristics. Furthermore, existing theoretical guarantees do not always closely match what is observed in practice. Thus, we are dealing with an open problem that is very important to extend and solve.

The broad goal of the research presented in this thesis can be nicely stated in terms of Diagram (1.1). That is, we wish to extend learning theory and algorithms to consider both of the important illustrated stages. We will see that designing and providing guarantees for algorithms that help automatically select a kernel function is an important way to address this task.

1.2 Learning Scenario

In this section, we introduce terminology and notation that will be used throughout the thesis. To begin with, the goal of *supervised learning* is to select a hypothesis h from a set of hypotheses H that, when given an instance x from the set of instances $\mathcal{X}$, can accurately predict an associated label y from a set of feasible labels $\mathcal{Y}$. For example, if the task at hand is spam-detection, the set $\mathcal{X}$ will represent the set of all email messages and $\mathcal{Y} = \{\text{spam}, \text{non-spam}\}$ is the set of feasible labels.

As mentioned in the previous section, a hypothesis h does not operate directly on an instance, but rather on *features* derived from the instance. The explicit features of an instance x, are denoted by the vector $\Phi(x) \in \mathbb{R}^n$. Thus, $\Phi : \mathcal{X} \to \mathbb{R}^n$ is a *feature mapping* and defines the choice of features. Returning to the spam-detection example, a useful feature vector may be the number of times n distinct keywords appear within an email.

The notion of accuracy will depend on the *loss function*, which will depend on the task. In the *classification* setting we generally have $\mathcal{Y} = \{+1, -1\}$ and the loss function of interest is the zero-one loss. Given an instance x with associated label y, the zero-one loss of a hypothesis $h : \mathbb{R}^n \to \mathbb{R}$ is,

$$L : \mathbb{R} \times \mathcal{Y} \to \{0, 1\} \tag{1.2}$$

$$(h(x), y) \mapsto \mathbf{1}_{\{\text{sign}(h(x)) \neq y\}} \tag{1.3}$$

$$= \mathbf{1}_{\{h(x)y<0\}} \ , \tag{1.4}$$

where $\mathbf{1}_\omega$ is the indicator function of the event ω. Additionally, the margin-loss is also useful in analyzing an algorithm's performance,

$$L_\rho : \mathbb{R} \times \mathcal{Y} \to \{0, 1\} \tag{1.5}$$

$$(h(x), y) \mapsto \mathbf{1}_{\{h(x)y<\rho\}}\,. \tag{1.6}$$

Here, the hypothesis must not only predict the correct sign, but also with a margin of at least ρ. In the *regression* setting, we usually have $\mathcal{Y} \subseteq \mathbb{R}$ and the goal is not to predict the label exactly, but only to get "close". One of the most common losses for this setting is the squared loss,

$$L_2 : \mathbb{R} \times \mathcal{Y} \to \mathbb{R} \tag{1.7}$$

$$(h(x), y) \mapsto (h(x) - y)^2\,. \tag{1.8}$$

It is assumed that instances arrive according to a fixed and unknown distribution. Thus, the goal is to minimize the true error (or *risk*) according to this distribution,

$$R(h) = \operatorname*{E}_x[L(h(x), y)]\,. \tag{1.9}$$

Since we do not know the underlying distribution nor do we know the labels of all instances, directly measuring $R(h)$ is not possible. Instead, we are given a labeled training sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$ and can use it to measure the *empirical error*,

$$\widehat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} L(h(x_i), y_i) \,. \tag{1.10}$$

A very important point to realize is that fitting perfectly to the training set, so that $\widehat{R}(h) = 0$, is not enough to guarantee that $R(h)$ will be small. In fact, in such cases we may be *over-fitting* to the training data. Thus, in order to empirically estimate the true performance of a hypothesis, a portion of the labeled data is left aside and not used during training. The error on this *test set* is reported as the *test error*. A third set of data, called a *validation set*, is sometimes also left aside in order to tune the performance of certain parameters of the learning algorithm, before evaluating its performance of the final hypothesis on the test set.

Furthermore, we can relate the true and empirical error via theoretical guarantees known as *generalization bounds*. A generalization bound takes the following form,

$$R(h) \leq \widehat{R}(h) + C(m, H) \,, \tag{1.11}$$

where $C(m, H)$ is a *complexity* term which depends on the richness of the hypothesis class H and sample size m. Bounds of this type suggest two things: First, since the complexity term is expected to decrease with m, more data is desirable and will allow for a better estimate of the true error. Secondly, since the complexity term is expected to increase with the richness of the hypothesis class, there is a trade-off between improving the hypothesis class in order to reduce the empirical error while still not increasing the complexity term too much. That is we wish to avoid the negative effects of *over-fitting* to the training data.

Many modern learning algorithms, such as support vector machine and kernel ridge regression, make use of such bounds when selecting a hypothesis. That is, they select a hypothesis that makes a trade-off between minimizing the training error and limiting the complexity of the selected hypothesis:

$$\operatorname*{argmin}_{h \in H} \widehat{R}(h) + C(H)\,, \tag{1.12}$$

where here the number of training points m has been fixed. Progressively more complex hypothesis sets may be considered until the best trade-off between training error and complexity is found. Such an approach is known as *structural risk minimization* (SRM) (Vapnik & Chervonenkis, 1974).

1.3 Kernel Methods

Kernel methods have been successfully used in a variety of learning tasks with the best known example of support vector machines (SVMs) (Boser et al., 1992; Cortes & Vapnik, 1995; Vapnik, 1998). In this section we briefly introduce the concept of kernels as well as some example algorithms. For a much more thorough treatment of the topic, the reader is directed to Schölkopf and Smola (2002) or Shawe-Taylor and Cristianini (2004).

As the name would suggest, kernel methods or kernel based algorithms, depend on a *kernel function.* A kernel function is a similarity measure,

$$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \tag{1.13}$$

$$(x, x') \mapsto K(x, x')\,, \tag{1.14}$$

that returns a real value characterizing the similarity of x and x' (Schölkopf & Smola, 2003). For example, if $\mathcal{X} = \mathbb{R}^n$, then the standard dot product provides a reasonable similarity measure. We will focus on functions that implicitly define a dot-product in a *feature space*, which may be different than $\mathcal{X}$, and can be characterized via a simple criteria as explained below.

Definition 1.1 ((Positive Definite) Kernel Function)**.** *A* positive definite kernel function *is a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that defines a dot product in a reproducing kernel Hilbert space $\mathcal{H}_K$, defined by the mapping $\Phi : \mathcal{X} \to \mathcal{H}_K$,*

$$\forall x, x' \in \mathcal{X},\ \ K(x, x') = \langle \Phi(x), \Phi(x') \rangle\,. \tag{1.15}$$

Furthermore, any function that is symmetric, i.e.

$$K(x, x') = K(x', x)\,, \tag{1.16}$$

and positive semi-definite, i.e. $\forall N \in \mathbb{N}, \forall c \in \mathbb{R}^N, \forall x \in \mathcal{X}^n$

$$\sum_{i,j=1}^{N} c_i c_j K(x_i, x_j) \geq 0\,, \tag{1.17}$$

is necessarily a (positive definite) kernel function.

Throughout this thesis, for brevity, we use the term kernel synonymously with positive definite kernel. It will also be useful to define the kernel *matrix* which corresponds to a kernel function evaluated on a sample.

Definition 1.2 (Kernel Matrix)**.** *A* kernel matrix *or* Gram matrix *associated to a kernel K and sample $S = (x_1, \ldots, x_m) \in \mathcal{X}^m$, is the matrix $\mathbf{K}$ defined as follows:*

$$\mathbf{K}_{ij} = K(x_i, x_j)\,. \tag{1.18}$$

Note that, by the definition of the kernel function, the kernel matrix is symmetric and positive semi-definite or, equivalently, its eigenvalues are all non-negative.

The characterization of a kernel function is relatively general, allowing for a wide variety of kernels. Some examples include:

Polynomial: Given instances $\mathbf{x}, \mathbf{x}' \in \mathcal{X} \subseteq \mathbb{R}^N$, the polynomial kernel is defined as follows,

$$K_{p,c}(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}' + c)^p\,, \tag{1.19}$$

where the parameter c allows for a constant offset and $p > 0$ controls the degree of the polynomial. In this case, the explicit feature map Φ can be constructed by mapping each vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ to a vector which considers all p tuples. For example, in the case of $p = 2$ and $c = 0$, we have $\Phi(\mathbf{x}) = (x_1 x_1, \sqrt{2} x_1 x_2, \ldots, x_n x_n) \in \mathbb{R}^{N^2}$.

Gaussian: Given instances $\mathbf{x}, \mathbf{x}' \in \mathcal{X} \subseteq \mathbb{R}^N$, the Gaussian kernel is defined as follows,

$$K_\sigma(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{\sigma^2}\right) , \tag{1.20}$$

where the *bandwidth* parameter $\sigma > 0$ controls the sensitivity of the kernel function. It can be shown that the feature map Φ associated to the Gaussian kernel actually corresponds to a space with *infinite* dimension (Schölkopf and Smola (2002), Remark 12.37).

Sequence Based: Kernel functions do not necessarily need to operate only on vectors in $\mathbb{R}^N$. For example, if we define the set $\mathcal{X}$ as all variable length sequences with a finite alphabet, then kernels can be constructed from objects such as weighted transducers, which are similar to weighted automata but with both input and output labels. Such kernels are known as *rational kernels*. If we denote a weighted transducer as T, then, for all $x, x' \in \mathcal{X}$, a valid sequence-based kernel can take the form,

$$K_T(x, x') = T \circ T^{-1} , \tag{1.21}$$

where $\circ$ denotes the transducer composition operation and T^{-1} is the transducer T with input and output labels interchanged. For a detailed description of these operations and rational kernel in general, the reader should reference Cortes et al. (2004).

It may be evident from these examples that the mapping Φ need not be linear. Thus, a linear separator in the induced feature space corresponds to a non-linear separation in the input space. For this reason, one may view kernels as a way to transform a linear algorithm into a non-linear one.

If the value of dot-products between instances in features space is all that is required from the data, then kernels have the advantage of computing these dot-products implicitly. That is, we can compute the dot-product without having to actually map the instance into the feature space via the function Φ. This will allow for better efficiency in cases where Φ is a mapping to a *very* high dimensional space. Using a kernel function also allows for flexibility, since the choice of features can be changed simply by changing the definition of the kernel function. However, is it ever the case that we only need the value of dot-products between data-points? In fact, this can often be the case as is exemplified by many “kernelized” algorithms. These are exactly algorithms which represent the training data only in terms of inner products between the points, in other words, algorithms which depend only on a kernel matrix in order to train.

Such algorithms appear within many tasks found in machine learning:

Classification:

Support Vector Machine – Perhaps the most well-known and most studied kernel-based algorithm is Support Vector Machines (SVM’s), which finds a maximum margin linear separator in the feature space induced by Φ (Boser et al., 1992; Cortes & Vapnik, 1995; Vapnik, 1998).

Kernel Logistic Regression – A maximum likelihood method, which assumes a logistic distribution over observations and allows non-linear separation when used with a kernel (Wahba et al., 1993).

Regression:

Kernel Ridge Regression – Used to find a regularized least-squares solution in kernel feature space (Saunders et al., 1998).

Support Vector Regression – An algorithm similar to the SVM problem, but with a regression-based loss (Drucker et al., 1997).

Clustering:

Kernel K-Means – An iterative clustering algorithm, which can separate clusters non-linearly with the use of a kernel function (Dhillon et al., 2004).

Maximum Margin Clustering – An algorithm that separates data into two clusters with maximum margin in the kernel feature space (Xu et al., 2005).

Dimensionality Reduction:

Kernel Principle Component Analysis – Kernels allow PCA, an unsupervised dimensionality reduction method, to map data into the desired lower-dimensional space in a non-linear fashion (Schölkopf et al., 1998).

Kernel Linear Discriminant Analysis – A supervised dimensionality reduction method that also allows for non-linear projections via kernel functions (Baudat & Anouar, 2000).

This list serves only as a very small sampling of kernel-based algorithms, but also illustrates the wide impact that kernel-based algorithms have made. However, in none of these algorithms is the proper choice of kernel taken into consideration and is left to the user. The proper choice of kernel will depend on the data and the particular task that is being addressed. In practice, the kernel is often selected in an ad-hoc fashion, by simply trying several different kernels and evaluating performance via cross-validation methods. Otherwise, the kernel function may be engineered specifically for the particular dataset, which is potentially time-consuming and will require domain knowledge (Schölkopf & Smola, 2002). Instead, in this thesis, we explore alternatives to this entirely manual selection of a kernel function for use with kernel algorithms. The next section gives an overview of existing work in this field.

1.4 Automatic Kernel Selection

The choice of the kernel is critical to the success of the algorithm. One way to see the importance of the kernel is from the fact that choosing a kernel is equivalent to choosing a feature space. As was shown in the previous section, using only a few examples, there are very different types of kernels available to the user. Furthermore, even for each kernel type there are several parameters to choose that will define the final kernel function.

A weaker commitment is required from the user when instead the kernel is *learned* from data. One can then specify a family of kernels, $\mathcal{K}$, and let a learning algorithm use the data to select both the kernel out of this family as well as the prediction hypothesis. For example, one general algorithm is to minimize a bound of the form,

$$\min_{K \in \mathcal{K}} \min_{h \in \mathcal{H}_K} \widehat{R}(h) + C(\mathcal{K}, \mathcal{H}_K) \tag{1.22}$$

which attempts to minimize the empirical error plus a complexity term that controls the richness of both choice of the kernel class and the hypothesis class.

In this section, we review the history of such methods, up to the state of the art theory and algorithms. The separate theoretical and algorithmic contributions of these previous results are explained in further detail in sections 2.1 and 3.1 respectively.

Some of the first work that can be considered automatic kernel selection focused on selecting from a relatively restricted class of kernels, such as the family of Gaussian kernels which are parametrized by the bandwidth parameter. Cristianini et al. (1999) use an iterative algorithm to minimize a bound of the type shown in (1.22) or the validation error directly in order to automatically tune the bandwidth parameter, σ, of the Gaussian kernel. Chapelle et al. (2002) also used bounds on the expected error of SVM as an objective to minimize in order to select multiple parameters of a multi-scale Gaussian kernel as well as the SVM trade-off parameter and optimized the function via gradient descent. The generalization bounds that were considered include the *radius-margin* bound, which considers the ratio of the radius of the data and the margin of the separating hyperplane, and the related *span-bound*. Such methods effectively allow the user to search a much larger set of kernel parameters than what is allowed with traditional cross-validation techniques, but is shown to have comparable performance. Similar types of bounds are optimized by Weston et al. (2001); Grandvalet and Canu (2003) in order to solve the different, but related, task of feature selection for SVM. Such a method allows for significant performance gain when it is known a priori that many features are non-informative.

Another line of research, initially presented by Cristianini et al. (2001), focuses on choosing the kernel in order to maximize a criteria that does not have a direct relationship with the learning algorithm or generalization bound, but which is rather considered to generally measure the quality of a kernel with respect to a training sample. That is, they suggest maximizing the *alignment* of the kernel with the training labels. The alignment function, which is discussed in much more detail in Section 3.3, is related to the correlation between the random variables $K(x, x')$ and yy', where y (respectively y') is the label corresponding to x (respectively x'). Here the authors consider decomposing a kernel matrix and then learning a weighting for each eigenvalue, which results in a new matrix with maximal alignment. Since in this scenario only the kernel *matrix*, and not kernel *function*, is being learned, we are restricted to the transductive setting, i.e. where the test points are known during training time.

Following this, Lanckriet et al. (2002) introduced what are now the most popular and widely studied family of kernels. That is, the family of kernels generated by linear combinations of fixed base kernels. The size of the family of kernels is controlled by restricting the L_1-norm of the combination weight vector. The base kernels are allowed to be general, so that they include, for example, Gaussian kernels with varying bandwidths or polynomial kernels of different degrees, but also allow general combinations of entirely different kernels generated from possibly entirely different sets of raw features. The final kernel is chosen by optimizing the SVM or KRR objective function directly and simultaneously while optimizing the standard learning parameters. Thus, this is also the first method to directly optimize the objective of the learning algorithm for which the kernel is used with. This optimization, over both sets of variables, is reduced to a semi-definite program, in the general case, or to a quadratically constrained quadratic program in the case all combination weights are constrained to be positive. Thus, in both cases the optimization task is shown to be convex and solvable in polynomial time. This paper also presented the first generalization bounds for this richer hypothesis class that also takes into account the family of linear combinations of base kernels. Similar bounds are given by Bousquet and Herrmann (2002) as well as a gradient descent style algorithm to solve similar optimization problems. Here too, the algorithms presented were for the transductive setting and again were for learning a kernel matrix. However, it should be noted that if the underlying kernel functions that generate the base kernels are known, the results are easily extended to learning the kernel function as well.

It was not until later that Srebro and Ben-David (2006) proved several of the previous generalization bounds, which depend on spectral properties of the base kernel matrices (Lanckriet et al., 2004a; Bousquet & Herrmann, 2002), were in fact vacuous. They then went on to give the first informative margin-based classification bounds that contained an additive term that measures the complexity of the linear kernel family. The only non-vacuous generalization bound up to that point contained a multiplicative factor (Lanckriet et al., 2004a). Finally, Cortes et al. (2009a) also gave additive generalization bounds that hold specifically for the kernel ridge regression algorithm. These bounds, as well as new state of the art bounds are introduced in Chapter 2.

In related theoretical work, Argyriou et al. (2005) consider the scenario of infinite convex combinations of kernels via a continuously parameterized kernel class. They show that when using such a family to optimize particular types of regularized loss functions, at most $m + 1$ kernels will have non-zero weights at the optimal solution, where m is the number of training points.

More complex families of kernels have also been proposed, such as those generated by *hyperkernels* (Ong et al., 2005). A hyperkernel is defined as a kernel function that operates on a Hilbert space, which itself contains kernel functions. An optimization problem can then be defined by choosing a function from this class of kernel functions which maximizes a certain quality functional, while regularizing by the norm of the function as measured by the hyperkernel. Non-linear combinations of base kernels have also been considered (Varma & Babu, 2009; Bach, 2008; Cortes et al., 2009b) in recent literature.

Automatically selecting useful features is a main motivation in the subfield of feature selection (Guyon & Elisseeff, 2003; Blum & Langley, 1997; Kohavi & John, 1997; Liu & Yu, 2005). The work presented in this thesis and the general subfield of learning with multiple kernels, however, differs from feature selection in several important aspects. As has been explained previously, with the use of kernels, explicit features are not needed and in fact may not be known. Also, in this work, the goal will not be to necessarily select an optimal subset of features, but rather learn the best set of weighted features. This allows for a strictly more general setting.

The focus of this thesis will be on learning with linear families of kernels, which are at the center of the wide majority of the automatic kernel selection theory as well as practical algorithms. The specific setting, problems and results are reviewed in the introduction to each chapter.

1.5 Contributions

We investigate several aspects of learning with multiple kernels, extending both theoretical foundations and algorithmic results.

When learning with multiple kernels, standard bounds on the complexity of the hypothesis class no longer hold and further analysis is needed in order to guide algorithmic development. Current state-of-the-art margin based generalization bounds have an additive dependence on the number of base kernels.

In our main theoretical result, we give a novel analysis of the Rademacher complexity of a hypotheses set generated using multiple kernels. This results in *tight* margin-based bounds for several families of linearly combined kernels. When considering convex combinations of kernels, there is in fact only a *logarithmic* dependence on the number of base kernels. This encourages the use of a very large number of base kernels, as long as it helps minimize an empirical margin-based loss. We also show the first generalization bounds for the regression setting, using a specialized stability analysis unlike in any previously shown bounds.

On the algorithmic side, we observe that it has been difficult in the past to always outperform a simple baseline *uniform* combination of base kernels. We show a modified kernel ridge regression algorithm (LKRR), which uses a non-sparse combination of base kernels, that in fact improves upon the performance of the uniform baseline and illustrates that previously suggested sparse combinations of kernels are not always beneficial.

We also give a modified definition of the alignment measure introduced by Cristianini et al. (2001). This measure is useful in assessing the quality of a kernel independent of any specific learning algorithm. This new definition addresses an important problem that is exhibited both with artificial and real-world data. We show simpler and novel concentration bounds that directly measure the difference between the true alignment and the empirical estimate, and which shows that the alignment can be measured from samples.

Then, using this newly introduced alignment measure, we introduce two-stage algorithms where the kernel is selected separately from the learned hypothesis. In experiments with these algorithms, we use general base kernels that, to the best of our knowledge, had not been investigated empirically before. In several settings, this method shows improvement over the uniform baseline, as well as more complicated one-stage methods.

The algorithms presented in this thesis have also been implemented in the open-source library OpenKernel (Allauzen et al., 2010).

Chapter 2

Theoretical Foundations

2.1 Previous Results

This section presents several novel generalization bounds for the problem of learning kernels for the family of non-negative combinations of base kernels with an L_1 or L_2 constraint. That is, for the problem of selecting a kernel from the family

$$\mathcal{K} = \left\{ \sum_{k=1}^{p} \mu_k K_k : \boldsymbol{\mu} \succeq 0, \|\boldsymbol{\mu}\|_q^q \leq \Lambda \right\}, \tag{2.1}$$

usually for $q \in \{1, 2\}$, but also more generally for some $q \geq 1$.

Recall from Section 1.2, a standard generalization bound emphasizes a trade-off between the empirical error and the complexity of the hypothesis class. By enriching our hypothesis class to consider not only a fixed kernel, but a family of kernels, we expect the empirical error to improve. Our concern, however, should be whether the complexity of the hypothesis class has increased too much, and if we are now susceptible to over-fitting. As can be expected, the complexity of the kernel family will grow with p, the number of base kernels. The degree to which the complexity depends on p will be the focus of the bounds that are discussed.

One of the first learning bounds given by Lanckriet et al. (2004a) for the family of convex combinations of p base kernels with an L_1 constraint is similar to that of Bousquet and Herrmann (2002) and has the following form:

$$R(h) \leq \widehat{R}_\rho(h) + O\left(\frac{1}{\sqrt{m}}\sqrt{\max_{k=1}^{p} \operatorname{Tr}(\mathbf{K}_k) \max_{i=1}^{p}(\|\mathbf{K}_k\|/\operatorname{Tr}(\mathbf{K}_k))/\rho^2}\right). \quad (2.2)$$

where $R(h) = \Pr[yh(x) < 0]$ is the generalization error of a hypothesis h, $\widehat{R}_\rho(h) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{y_i h(x_i) < \rho}$ is the fraction of training points with margin less than ρ, and $\mathbf{K}_k$ is the kernel matrix associated to the kth base kernel. This bound was later shown by Srebro and Ben-David (2006) to be always larger than one. Another bound by Lanckriet et al. (2004a) for the family of linear (not necessarily convex) combinations of kernels was also shown, by the same authors, to be always larger than one.

However, by considering a sum over K_k instead of a $\max_k$, Lanckriet et al. (2004a) also presented a multiplicative bound for convex combinations of base kernels with an L_1 constraint that is of the form

$$R(h) \leq \widehat{R}_\rho(h) + O\left(\sqrt{\frac{pR^2/\rho^2}{m}}\right), \quad (2.3)$$

where R is a bound on the kernel function, $\sup_{x \in \mathcal{X}} K(x,x) \leq R^2$. This bound converges and can perhaps be viewed as the first informative generalization bound for this family of kernels. However, the dependence of the bound on the number of kernels p is multiplicative and therefore does not encourage the use of too many base kernels. However, this does not seem to capture the behavior of algorithms that use multiple kernels in practice. This is evident in experiments where $p \approx m$ and there is no apparent effect of overfitting (for example in Section 3.2 and 3.3). Srebro and Ben-David (2006) presented a generalization bound based on the pseudo-dimension of the family of kernels that significantly improved on this bound. Their bound has the form

$$R(h) \leq \widehat{R}_\rho(h) + \widetilde{O}\Big(\sqrt{\frac{p + R^2/\rho^2}{m}}\Big), \tag{2.4}$$

where the notation $\widetilde{O}(\cdot)$ hides logarithmic terms and where R^2 is an upper bound on $K_k(x,x)$ for all points x and base kernels K_k, $k \in [1,p]$. Thus, disregarding logarithmic terms, their bound is only additive in p. Their analysis also applies to other families of kernels. Ying and Campbell (2009) also gave generalization bounds for learning kernels based on the notion of Rademacher chaos complexity and the pseudo-dimension of the family of kernels used. For a pseudo-dimension of p as in the case of a convex combination of p base kernels, their bound is in $O(\sqrt{p\,(R^2/\rho^2)(\log(m)/m)})$ and is thus multiplicative in p. It seems to be weaker than the bound of Lanckriet et al. (2004a) and that of Srebro and Ben-David (2006) for such kernel families.

2.2 Novel Generalization Bounds

In Section 2.3 we prove the first known stability-based bound for the problem of learning the kernel. This bound holds for a kernel ridge regression type algorithm that considers non-negative linear combinations of kernels with an L_2 type constraint on combination the weights. Thus, to the best of our knowledge, it is also the first regression bound for learning with multiple kernels. Since the bound is specifically for regression, it is not directly comparable to the previous margin based bounds. However, when compared to the standard stability bound with a fixed kernel, this more general bound has only an additional additive term of the form $O(\sqrt{p/m})$. The proof of this bound provides novel techniques for providing stability bounds and can be used to give even tighter bounds in the standard fixed kernel setting.

In Section 2.4 we prove tight margin bounds for both L_1 and L_2, as well as more general L_q regularized non-negative combinations of kernels. These bounds demonstrate that generalization is still possible even with relatively large numbers of kernels (even if $p > m$). In particular, Corollary 2.1 gives margin-based bound of the following form for hypotheses based on convex combination of kernels,

$$R(h) \leq \widehat{R}_\rho(h) + O\Big(\sqrt{\frac{\log(p)R^2/\rho^2}{m}}\Big)\,. \tag{2.5}$$

If we consider the constants as well, the corollary gives a complexity term for learning convex combinations of kernels that is

$$2\sqrt{\frac{\eta_0 e \lceil \log p \rceil R^2/\rho^2}{m}},$$

where $\eta_0 = \frac{23}{22}$. In comparison, the best previous complexity bound for learning kernels with convex combinations given by Srebro and Ben-David (2006) derived using the pseudo-dimension has a stronger dependency with respect to p and is more complex:

$$\sqrt{8\frac{2 + p\log\frac{128em^3R^2}{\rho^2 p} + 256\frac{R^2}{\rho^2}\log\frac{\rho em}{8R}\log\frac{128mR^2}{\rho^2}}{m}}.$$

Note, this bound is also not informative for $p > m$.

Figure 2.1 compares the bound on $R(h) - \widehat{R}_\rho(h)$ obtained using this expression by Srebro and Ben-David with the new bound shown in Equation (2.5), as a function of the sample size m. The comparison is made for different values of the number of kernels p, a normalized margin of $\rho/R = .2$ and the confidence parameter set to $\delta = .01$ (see Corollary 2.1 for the minor $\log\frac{1}{\delta}$ dependence). Plots for different values of the normalized margin are quite similar. As shown by the figure, larger values of p can significantly affect the bound of Srebro and Ben-David leading to quasi-flat plots for $p > m^{4/5}$. In comparison, the plots for our new bound show only a mild variation with p even for relatively large values such as $p \sim m$. Note also that, while the bound of Srebro and Ben-David does converge and becomes informative, its values,

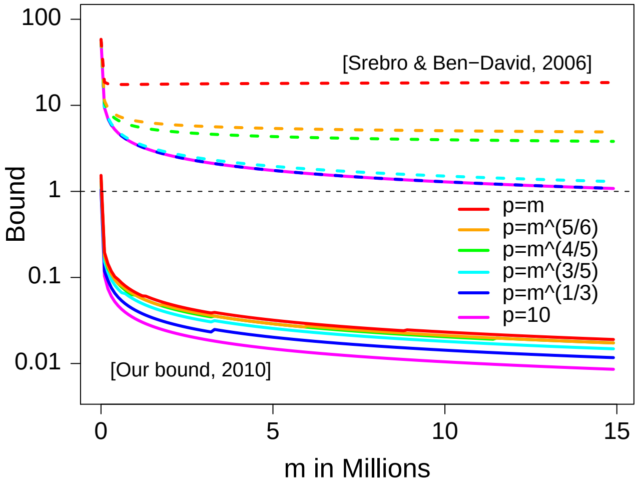
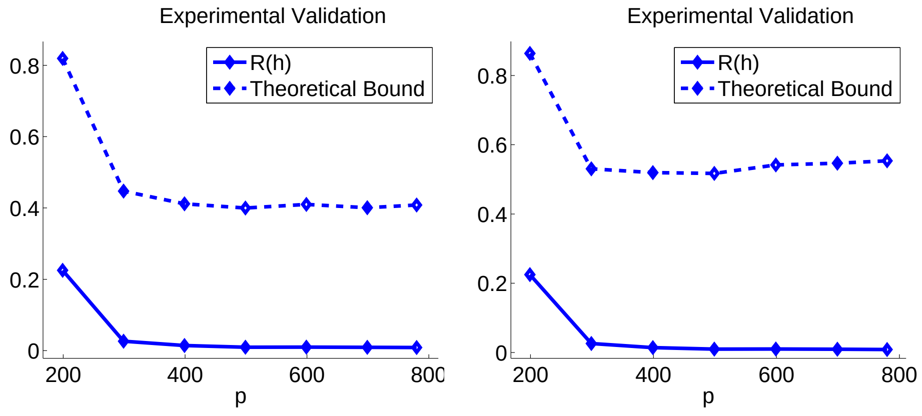


Figure 2.1: Plots of the bound of Srebro & Ben-David (dashed lines) and our new bounds (solid lines) as a function of the sample size m for $\delta = .01$ and $\rho/R = .2$. For these values and $m \leq 15 \times 10^6$, the bound of Srebro and Ben-David is always above 1, it is of course converging for sufficiently large m. The plots for $p = 10$ and $p = m^{1/3}$ roughly coincide in the case of the bound of Srebro & Ben-David, which makes the first one not visible.

even for $p = 10$, are still above 1 for fairly large values of m. The new bound, in contrast, strongly encourages considering large numbers of base kernels in learning kernels.

The $\sqrt{\log p}$ dependency of our generalization bound with respect to p cannot be improved upon. This can be seen by arguments in connection with the VC dimension lower bounds. Consider the case where the input space is $\mathcal{X} = \{\text{-}1, +1\}^p$ and where the feature mapping of each base kernel K_k, $k \in [1, p]$, is simply the canonical projection $\mathbf{x} \mapsto +x_k$ or $\mathbf{x} \mapsto -x_k$, where x_k is the kth component of $\mathbf{x} \in \mathcal{X}$. Thus, H_1^p then contains the hypothesis set $J^p = \{\mathbf{x} \mapsto sx_k \colon k \in [1, p], s \in \{\text{-}1, +1\}\}$ whose VC dimension is in $\Omega(\log p)$. For $\rho = 1$ and $h \in J^p$, for any $x_i \in \mathcal{X}$, $y_i h(x_i) < \rho$ is equivalent to $y_i h(x_i) < 0$.



(a) L_1 Bound (b) L_2 Bound

Figure 2.2: Comparison of the behavior of the experimentally determined test error as a function of the number of kernels, versus that of the bound on $R(h)$ given by (a) Corollary 2.1 for experiments with L_1 regularization, and by (b) Corollary 2.2 for L_2 regularization. In these examples $m = 36{,}000$, the normalized margin is $\rho/R = .2$, and the confidence parameter δ is set to .01.

Thus, the empirical margin loss $\widehat{R}_\rho(h)$ coincides with the standard empirical error $\widehat{R}(h)$ for $h \in J^p$ and a margin bound with $\rho = 1$ implies a standard generalization bound with the same complexity term. By the classical VC dimension lower bounds (Devroye et al., 1996; Anthony & Bartlett, 1999), that complexity term must be at least in $\Omega\big(\sqrt{\mathrm{VCDim}(J^p)/m}\big) = \Omega(\sqrt{\log p/m})$.

We have also tested experimentally the behavior of the test error as a function of p and compared it to that of the theoretical bound given by Equation (2.5) by learning with a large number of kernels $p \in [200, 800]$, a sample size of $m = 36{,}000$, and a normalized margin of $\rho/R = .2$. These results are for rank-1 base kernels generated from individual features of the MNIST dataset (Lecun & Cortes, 1998). The magnitude of each kernel weight is chosen proportionally to the correlation of the corresponding feature with the training labels. The results show that the behavior of the test error as a function of p matches the one predicted by our bound, see Figure 2.2(a).

We note that Koltchinskii and Yuan (2008) also presented a bound with logarithmic dependence on p in the context of the study of large ensembles of kernel machines. However, their analysis is specific to the family of kernel-based regularization algorithms and requires the loss function to be strongly convex, which rules out for example the binary classification loss function. Also, both the statement of the result and the proof seem to be considerably more complicated than ours.

We are also able to provide bounds for L_q regularized combinations. These bounds hold for any q, such that $\frac{1}{q}+\frac{1}{r}=1$ and $r>1$ is an integer. Corollary 2.2 gives a bound of the form

$$R(h) \leq \widehat{R}_\rho(h) + O\bigg(\sqrt{\frac{rp^{1/r}R^2/\rho^2}{m}}\bigg). \tag{2.6}$$

In particular, for $q=2$, the bound has a multiplicative dependence of $p^{1/4}$.

Figure 2.3 shows a comparison of the L_2 regularization bound of Equation (2.6) with the L_1 regularization bound of Equation (2.5). As can be seen from the plots, the two bounds are very close for smaller values of p. For larger values ($p \sim m$), the difference becomes significant. The bound for L_2 regularization is converging for these values but at a slower rate of $O\big(\frac{R/\rho}{m^{1/4}}\big)$.

As with the L_1 bound we also tested experimentally the behavior of the

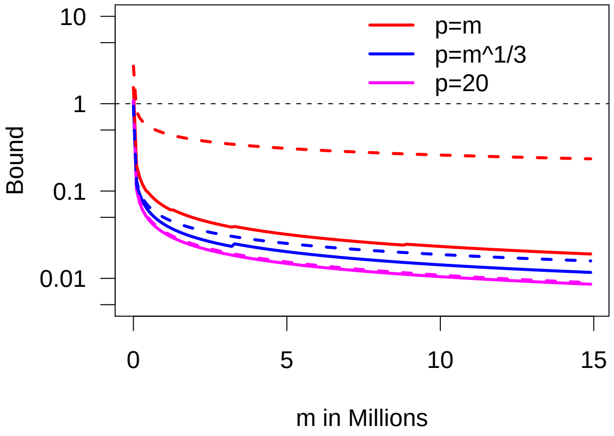


Figure 2.3: Comparison of the L_1 regularization bound of Corollary 2.1 and the L_2 regularization bound of Corollary 2.2 (dotted lines) as a function of the sample size m for $\delta = .01$ and $\rho/R = .2$. For $p = 20$, the L_1 and L_2 bounds roughly coincide.

test error as a function of p and compared it to that of the theoretical bound given by Equation (2.6) by learning with a large number of kernels. Again, our results show that the behavior of the test error as a function of p matches the one predicted by our bound, see Figure 2.2(b). The $p^{1/(2r)}$ dependency of the generalization bound of Equation (2.6) also cannot be improved. This holds, for example, when the base kernels are all equal and is shown explicitly in Section 2.4.

The analysis for these bounds provides a novel analysis for bounding the Rademacher complexity of kernel based hypothesis classes. This analysis also leads to improvements in the standard fixed kernel setting.

In the following sections, we prove the results that were discussed in this section, showing both stability-based as well as Rademacher-based proofs. Finally, in Section 2.5 we analyze theoretical properties of an alignment measure which can be used to evaluate the usefulness of a kernel independent of any particular learning algorithm. This quality measure will form the basis of the two-stage algorithms which are explored in Section 3.3.

2.3 Stability-Based Proofs

In this section, we derive generalization bounds for the LKRR algorithm, defined in Section 2.3.1, using the notion of algorithmic stability (Bousquet & Elisseeff, 2002). Our analysis focuses on the regression setting also examined by Micchelli and Pontil (2005) and Argyriou et al. (2005). More specifically, we will consider the problem of learning kernels in kernel ridge regression, KRR, (Saunders et al., 1998). The bounds we derive here can be considered the first regression bounds for this setting with only an *additive* dependence on the number of base kernels used, p. This is similar to what is presented by Srebro and Ben-David (2006) in the classification setting, however here we have no additional logarithmic terms. It is also the first algorithm specific analysis of a kernel selection algorithm, which is required when using algorithmic stability.

Definition 2.1 (Uniform β-Stable). *A learning algorithm is said to be (uniformly) β-*stable *if the hypotheses h' and h it returns for any two training samples, S and S', that differ by a single point satisfy*

$$\left|[h'(x) - y]^2 - [h(x) - y]^2\right| \leq \beta \tag{2.7}$$

for any point $x \in \mathcal{X}$ labeled with $y \in \mathbb{R}$.

The stability coefficient β is a function of the sample size m. Intuitively, if a function is stable, then the more training points that are used, the less the function h and h' should differ. That is, β is a decreasing function of m. Stability in conjunction with McDiarmid's inequality can lead to tight generalization bounds specific for the algorithm analyzed (Bousquet & Elisseeff, 2002).

In what follows we first introduce the optimization problem that is solved by LKRR, as well as the form of the solution. Using specific properties of this solution, we then analyze the stability of LKRR.

2.3.1 LKRR Optimization Problem

Let $S = ((x_1, y_1), \ldots, (x_m, y_m))$ denote the training sample and $\mathbf{y} = [y_1, \ldots, y_m]^\top$ the vector of training set labels, where $(x_i, y_i) \in \mathcal{X} \times \mathbb{R}$ for $i \in [1, m]$, and let $\Phi(x)$ denote the feature vector associated to $x \in \mathcal{X}$. Then, in the primal, the KRR optimization problem has the following form

$$\min_{\mathbf{w}} \|\mathbf{w}\|^2 + \frac{C}{m} \sum_{i=1}^{m} (\mathbf{w}^\top \Phi(x_i) - y_i)^2, \tag{2.8}$$

where $C \geq 0$ is a trade-off parameter. For a fixed positive definite kernel (PDS) function $K \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, the dual of the KRR optimization problem

(Saunders et al., 1998) is given by:

$$\max_{\boldsymbol{\alpha}} -\lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha} - \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{y}, \tag{2.9}$$

where $\lambda = m/C$. In the following, we will denote by λ_0 the inverse of C, thus, $\lambda = \lambda_0 m$.

Here, we limit the search to kernels K that are non-negative combinations of p fixed PDS kernels K_k, $k \in [1, p]$, and that are thereby guaranteed to be PDS, with an L_2 regularization:

$$\mathcal{K} = \{\sum_{k=1}^{p} \mu_k K_k \colon \boldsymbol{\mu} \in \mathcal{M}\}\,, \tag{2.10}$$

where,

$$\mathcal{M} = \{\boldsymbol{\mu} \colon \boldsymbol{\mu} \geq 0 \wedge \|\boldsymbol{\mu} - \boldsymbol{\mu}_0\|^2 \leq \Lambda^2\}\,, \tag{2.11}$$

with $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_p]^\top$, $\boldsymbol{\mu}_0 \succeq 0$ a fixed combination vector, and $\Lambda \geq 0$ a regularization parameter. The parameter $\boldsymbol{\mu}_0$ can be used to encode any prior knowledge of a “good” weighting. Alternatively it can be set to $\mathbf{0}$, in which case we have the standard L_2 regularization.

Based on the dual form of the optimization problem for KRR, the kernel learning optimization problem can be formulated as follows:

$$\min_{\boldsymbol{\mu} \in \mathcal{M}} \max_{\boldsymbol{\alpha}} \; -\lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha} - \underbrace{\sum_{k=1}^{p} \mu_k \boldsymbol{\alpha}^\top \mathbf{K}_k \boldsymbol{\alpha}}_{\boldsymbol{\mu}^\top \mathbf{v}} + 2\boldsymbol{\alpha}^\top \mathbf{y}, \tag{2.12}$$

where $\mathbf{K}_k$ is the Gram matrix associated to the base kernel K_k. It is convenient to introduce the vector $\mathbf{v} = [v_1, \ldots, v_p]^\top$ where $v_k = \boldsymbol{\alpha}^\top \mathbf{K}_k \boldsymbol{\alpha}$. Note that this defines a convex optimization problem in $\boldsymbol{\mu}$, since the objective function is linear in $\boldsymbol{\mu}$ and the pointwise maximum over $\boldsymbol{\alpha}$ preserves convexity, and since $\mathcal{M}$ is a convex set. We refer to this learning kernel KRR procedure as LKRR and denote by h the hypothesis it returns defined by $h(x) = \sum_{i=1}^m \alpha_i K(x_i, x)$ for all $x \in \mathcal{X}$, when trained on the sample S, where K denotes the PDS kernel $K = \sum_{k=1}^p \mu_k K_k$.

Form of the Solution

Theorem 2.1. *The solution $\boldsymbol{\mu}$ of the optimization problem (2.12) is given by*

$$\boldsymbol{\mu} = \boldsymbol{\mu}_0 + \Lambda \frac{\mathbf{v}}{\|\mathbf{v}\|} \tag{2.13}$$

with $\boldsymbol{\alpha}$ the unique vector verifying $\boldsymbol{\alpha} = (\sum_{k=1}^p \mu_k \mathbf{K}_k + \lambda \mathbf{I})^{-1} \mathbf{y}$.

Proof. By von Neumann's (1937) generalized minimax theorem, (2.12) is equivalent to its max-min analogue:

$$\max_{\boldsymbol{\alpha}} -\lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{y} + \min_{\boldsymbol{\mu} \in \mathcal{M}} -\boldsymbol{\mu}^\top \mathbf{v}, \tag{2.14}$$

where $\mathbf{v} = (\boldsymbol{\alpha}^\top K_1 \boldsymbol{\alpha}, \ldots, \boldsymbol{\alpha}^\top K_p \boldsymbol{\alpha})^\top$. The Lagrangian of the minimization problem is

$$L = -\boldsymbol{\mu}^\top (\mathbf{v} + \boldsymbol{\beta}) + \gamma (\|\boldsymbol{\mu} - \boldsymbol{\mu}_0\|^2 - \Lambda^2) \tag{2.15}$$

with $\boldsymbol{\beta} \geq 0$ and $\gamma \geq 0$ and the KKT conditions are

$$\nabla_{\boldsymbol{\mu}} L = -(\mathbf{v} + \boldsymbol{\beta}) + 2\gamma(\boldsymbol{\mu} - \boldsymbol{\mu}_0) = 0 \tag{2.16}$$

$$\nabla_{\boldsymbol{\beta}} L = \boldsymbol{\mu}^\top \boldsymbol{\beta} = 0 \Rightarrow (\frac{\mathbf{v} + \boldsymbol{\beta}}{2\gamma} + \boldsymbol{\mu}_0)^\top \boldsymbol{\beta} = 0 \tag{2.17}$$

$$\gamma(\|\boldsymbol{\mu} - \boldsymbol{\mu}_0\|^2 - \Lambda^2) = 0. \tag{2.18}$$

Note that if $\gamma = 0$ then the L_2 constraint is not met as an equality, which cannot hold at the optimum. By inspecting (2.12), it is clear that the μ_ks would be chosen as large as possible. Thus, the first equality implies $\boldsymbol{\mu} - \boldsymbol{\mu}_0 = \frac{\mathbf{v}+\boldsymbol{\beta}}{2\gamma}$, in view of which the second gives $-\|\boldsymbol{\beta}\|^2 = (\frac{\mathbf{v}}{2\gamma} + \boldsymbol{\mu}_0)^\top \boldsymbol{\beta}$. Since $\mathbf{v} \geq 0, \boldsymbol{\mu}_0 \geq 0, \gamma \geq 0$ and $\boldsymbol{\beta} \geq 0$, $(\frac{\mathbf{v}}{2\gamma} + \boldsymbol{\mu}_0)^\top \boldsymbol{\beta}$ is non-negative, which implies $-\|\boldsymbol{\beta}\|^2 \geq 0$ and $\boldsymbol{\beta} = 0$. The third equality gives $\boldsymbol{\mu} - \boldsymbol{\mu}_0 = \Lambda \frac{\mathbf{v}}{\|\mathbf{v}\|}$. Problem 2.14 can thus be rewritten as

$$\max_{\boldsymbol{\alpha}} \; \underbrace{-\lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{y} - \boldsymbol{\mu}_0^\top \mathbf{v}}_{\text{standard KRR with } \boldsymbol{\mu}_0\text{-kernel } \mathbf{K}_0.} - \Lambda \|\mathbf{v}\|. \tag{2.19}$$

For $\mathbf{v} \neq 0$, $\nabla_{\boldsymbol{\alpha}} \|\mathbf{v}\| = 2 \sum_{k=1}^p \frac{v_k}{\|\mathbf{v}\|} \mathbf{K}_k \boldsymbol{\alpha}$. Thus, differentiating and setting to zero the objective function of this optimization problem gives $\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y}$, with $\mathbf{K} = \sum_{k=1}^p \Big(\boldsymbol{\mu}_{0k} + \Lambda \frac{v_k}{\|\mathbf{v}\|}_{\mu_k}\Big) \mathbf{K}_k = \sum_{k=1}^p \mu_k \mathbf{K}_k$. □

2.3.2 Stability of LKRR

To analyze the stability of LKRR we consider two samples of size m, $S = (x_1, \ldots, x_m)$ and $S' = (x'_1, \ldots, x'_m)$, and without loss of generality we assume that the two samples differ only in the final point, x_m and x'_m. The task is then to bound the difference $|h'(x) - h(x)|$. The analysis is quite complex in this context and the standard convexity-based proofs of Bousquet and Elisseeff (2002) do not readily apply. This is because here, a change in a sample point also changes the PDS kernel K, which in the standard case is fixed.

Our proofs make use of the expression of $\boldsymbol{\alpha}$ and $\boldsymbol{\mu}$ supplied by Theorem 2.1. It is interesting to note, this analysis gives us a novel and tighter bound on the stability of standard KRR than the one obtained via convexity arguments (Bousquet & Elisseeff, 2002).

Fix $x \in \mathcal{X}$. We shall denote by $\Delta h(x)$ the difference $h'(x) - h(x)$ and more generally use the symbol Δ to abbreviate the difference between an expression depending on S' and one depending on S. We denote by $\mathbf{y}'$ the vector of labels, by K' the kernel learned by LKRR, and by μ'_k and $\boldsymbol{\mu}'$ the basis kernel coefficients and vector associated to the sample S'.

We will assume that the hypothesis set considered is bounded, that is $|h(x) - y(x)| \leq M$ for all $x \in \mathcal{X}$, for some $M \geq 0$. This bound and the Lipschitz property of the loss function implies a bound on $\Delta(h(x) - y)^2 \leq 2M\Delta h(x)$. We will also assume that the base kernels are bounded: there exists $R_0 \geq 0$ such that $(\sum_{k=1}^{p} K_k(x, x)^2)^{1/2} \leq R_0$ for all $x \in \mathcal{X}$. Thus, in conjunction with the regularization imposed on $\boldsymbol{\mu}$ in Equation (2.11), this implies that for all $x \in \mathcal{X}$, $K(x, x) = \sum_{k=1}^{p} \mu_k K_k(x, x) \leq R_0 \|\boldsymbol{\mu}\| \leq R_0(\|\boldsymbol{\mu}_0\| + \Lambda)$. Thus, we can assume that there exists $R \geq 0$ such that $K(x, x) \leq R^2$ for all $x \in \mathcal{X}$.

Now, $\Delta h(x)$ can be written as $\Delta h(x) = \Delta_S h(x) + \Delta_K h(x)$ to distinguish changes due to different samples (x_i's vs x_is) for a fixed kernel and those due to a different kernels K for a fixed sample:

$$\begin{aligned}
\Delta_S h(x) =& \sum_{i=1}^{m} \Big[(\sum_{k=1}^{p} \mu_k' \mathbf{K}_k(S') + \lambda \mathbf{I})^{-1} \mathbf{y}' \Big]_i \sum_{k=1}^{p} \mu_k' K_k(x_i', x) \\
& \qquad - \sum_{i=1}^{m} \Big[(\sum_{k=1}^{p} \mu_k' \mathbf{K}_k(S) + \lambda \mathbf{I})^{-1} \mathbf{y} \Big]_i \sum_{k=1}^{p} \mu_k' K_k(x_i, x), \\
\Delta_K h(x) =& \sum_{i=1}^{m} \Big[(\sum_{k=1}^{p} \mu_k' \mathbf{K}_k(S) + \lambda \mathbf{I})^{-1} \mathbf{y} \Big]_i \sum_{k=1}^{p} \mu_k' K_k(x_i, x) \\
& \qquad - \sum_{i=1}^{m} \Big[(\sum_{k=1}^{p} \mu_k \mathbf{K}_k(S) + \lambda \mathbf{I})^{-1} \mathbf{y} \Big]_i \sum_{k=1}^{p} \mu_k K_k(x_i, x).
\end{aligned}$$

Where $\mathbf{K}_k(S)$ (resp. $\mathbf{K}_k(S')$) is the kernel matrix generated from S (resp. S'). We bound these two terms separately. The main reason for this is that the term $\Delta_S h(x)$ leads to sparse expressions since the points x_is in S and S' differ only by x_m and x_m'. However, to bound $\Delta_K h(x)$ a different approach is needed. We will appropriately denote the stability coefficient of each term as β_S and β_K, and thus $|\Delta h(x)| \leq \beta_S + \beta_K$.

In what follows, we denote by Φ a feature mapping associated to kernel K and by $\mathbf{\Phi}$ the matrix whose columns are $\Phi(x_i)$, $i = 1, \ldots, m$. Similarly, for $k = 1, \ldots, p$, we denote by Φ_k a feature mapping associated with the base kernel K_k and by $\mathbf{\Phi}_k$ the matrix whose columns are $\Phi_k(x_i)$, $i = 1, \ldots, m$.

Bound on β_S

For the analysis of $\Delta_S h(x)$, the kernel coefficients μ'_k are fixed. Here, we denote by $\mathbf{K}$ the kernel matrix of $\sum_{k=1}^{p} \mu'_k \mathbf{K}_k$ over the sample S, and by $\mathbf{K}'$ the one over S'. Now, $h(x)$ can be expressed in terms of $\boldsymbol{\Phi}$ as follows:

$$
\begin{aligned}
h(x) &= [\boldsymbol{\Phi\alpha}]^\top \Phi(x) && (2.20)\\
&= \mathbf{y}^\top (\mathbf{K} + \lambda \mathbf{I})^{-1} \boldsymbol{\Phi}^\top \Phi(x) && (2.21)\\
&= \mathbf{y}^\top (\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \lambda \mathbf{I})^{-1} \boldsymbol{\Phi}^\top \Phi(x). && (2.22)
\end{aligned}
$$

Theorem 2.2. *Let $\lambda_{\min}(\mathbf{K}')$ denote the smallest eigenvalue of $\mathbf{K}'$. Then, the following bound holds for all $x \in \mathcal{X}$:*

$$
|\Delta_S h(x)| \leq \frac{2MR^2}{\lambda_{\min}(\mathbf{K}') + \lambda_0 m} \leq \beta_S \,. \tag{2.23}
$$

Proof. Using the general identity $(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \lambda \mathbf{I})^{-1} \boldsymbol{\Phi}^\top = \boldsymbol{\Phi}^\top (\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + \lambda \mathbf{I})^{-1}$, we can write equation (2.22) as

$$
h(x) = (\boldsymbol{\Phi}\mathbf{y})^\top (\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + \lambda \mathbf{I})^{-1} \Phi(x). \tag{2.24}
$$

Let $\mathbf{U} = (\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + \lambda \mathbf{I})$ and denote by $\mathbf{w}^\top$ the row vector $(\boldsymbol{\Phi}\mathbf{y})^\top \mathbf{U}^{-1}$. Now, we can write $\Delta_S h(x) = (\Delta_S \mathbf{w})^\top \Phi'(x)$. Using the identity $\Delta_S(\mathbf{U}^{-1}) = -\mathbf{U}^{-1}(\Delta_S \mathbf{U})\mathbf{U}'^{-1}$, valid for all invertible matrices $\mathbf{U}$ and $\mathbf{U}'$, $\Delta_S \mathbf{w}^\top$ can be expressed as follows:

$$\begin{aligned}\Delta_S \mathbf{w}^\top &= (\Delta_S \mathbf{\Phi y})^\top \mathbf{U}'^{-1} + (\mathbf{\Phi y})^\top \Delta_S(\mathbf{U}^{-1})\\ &= (\Delta_S \mathbf{\Phi y})^\top \mathbf{U}'^{-1} - (\mathbf{\Phi y})^\top \mathbf{U}^{-1}(\Delta_S \mathbf{U})\mathbf{U}'^{-1}.\end{aligned}$$

We observe that

$$(\Delta_S \mathbf{\Phi y}) = \Delta_S(\sum_{i=1}^m y_i \Phi(x_i)) = \sum_{i=1}^m (\Delta_S y_i \Phi(x_i)) = \Delta_S(y_m \Phi(x_m))$$

and

$$(\Delta_S \mathbf{U}) = \Delta_S(\sum_{i=1}^m \Phi(x_i)\Phi(x_i)^\top) = \Delta_S(\Phi(x_m)\Phi(x_m)^\top).$$

Thus, we can write

$$\begin{aligned}\Delta_S \mathbf{w}^\top &= \Big[\Delta_S(y_m\Phi(x_m))^\top - (\mathbf{\Phi y})^\top \mathbf{U}^{-1}\Delta_S(\Phi(x_m)\Phi(x_m)^\top)\Big]\mathbf{U}'^{-1}\\ &= \Big[y'_m \Phi(x'_m)^\top - y_m \Phi(x_m)^\top + (\mathbf{\Phi y})^\top \mathbf{U}^{-1}\Phi(x'_m)\Phi(x'_m)^\top\\ &\qquad\qquad - (\mathbf{\Phi y})^\top \mathbf{U}^{-1}\Phi(x_m)\Phi(x_m)^\top\Big]\mathbf{U}'^{-1}\\ &= \Big[(y'_m - h(x'_m))\Phi(x'_m) - (y_m - h(x_m))\Phi(x_m)\Big]^\top \mathbf{U}'^{-1}.\end{aligned}$$

Since for all $x \in \mathcal{X}$, $K(x,x) \le R^2$ and $|h(x) - y(x)| \le M$, we have $\|\Phi(x)\| \le R$ and $\|(y'_m - h(x'_m))\Phi(x'_m) - (y_m - h(x_m))\Phi(x_m)\| \le 2RM$, thus

$$\|\Delta_S \mathbf{w}^\top\| \le 2RM\|\mathbf{U}'^{-1}\|. \tag{2.25}$$

The smallest eigenvalue of $(\mathbf{\Phi}\mathbf{\Phi}^\top + \lambda\mathbf{I})$ is $\lambda_{\min}(\mathbf{\Phi}\mathbf{\Phi}^\top) + \lambda$. $\mathbf{\Phi}\mathbf{\Phi}^\top$ and $\mathbf{\Phi}^\top\mathbf{\Phi}$ have the same eigenvalues (the squares of the singular values of $\mathbf{\Phi}$). Thus, $\lambda_{\min}(\mathbf{\Phi}\mathbf{\Phi}^\top) = \lambda_{\min}(\mathbf{\Phi}^\top\mathbf{\Phi}) = \lambda_{\min}(\mathbf{K})$ and $\|\Delta_S \mathbf{w}^\top\| \leq \frac{2RM}{\lambda_{\min}(\mathbf{K}')+\lambda_0 m}$. Since $\|\Phi'(x)\| = K'(x,x) \leq R$, $|\Delta_S h(x)| \leq \frac{2R^2M}{\lambda_{\min}(\mathbf{K}')+\lambda_0 m}$. □

Recall, $\Delta_S h(x)$ represents the variation due to sample changes for a fixed kernel, thus, the bound given by the theorem is precisely a bound on the stability coefficient of standard KRR. This bound is tighter than the one obtained using the techniques of Bousquet and Elisseeff (2002): $|\Delta_S h(x)| \leq \frac{2R^2M}{\lambda_0 m}$.

Bound on β_K

Since $h(x) = \sum_{i=1}^m \alpha_i K(x_i, x)$, the variation in K can be decomposed into the following sum:

$$\Delta_K h(x) = \underbrace{\sum_{i=1}^m (\Delta_K \alpha_i) K'(x_i', x)}_{W} + \underbrace{\sum_{i=1}^m \alpha_i \Delta_K K(x_i', x)}_{T}.$$

By the Cauchy-Schwarz inequality, for any $x_i', x \in \mathcal{X}$,

$$|K(x_i', x)| \leq \sqrt{K(x_i', x_i')K(x, x)} \leq R^2,$$

thus the norm of the vector $k_{x'} = [K(x_1', x), \ldots, K(x_m', x)]$ is bounded by $R^2\sqrt{m}$ and the first term W can be bounded straightforwardly in terms of $\Delta_K \boldsymbol{\alpha}$: $|W| \leq R^2\sqrt{m}\|\Delta_K \boldsymbol{\alpha}\|$.

The second term can be written as follows

$$T = \sum_{i=1}^{m} \alpha_i \sum_{k=1}^{p} (\Delta\mu_k) K_k(x'_i, x) = \sum_{k=1}^{p} (\Delta\mu_k) (\mathbf{\Phi}_k \boldsymbol{\alpha})^\top \Phi_k(x). \tag{2.26}$$

By Lemma A.1 (see Appendix), $\Delta\mu_k$ can be expressed in terms of the Δv_ks and thus T can be rewritten as

$$T = \underbrace{\Lambda \sum_{k=1}^{p} \left[\frac{\Delta v_k}{\|\mathbf{v}'\|} - \frac{v_k \sum_{i=1}^{p} (v_i + v'_i) \Delta v_i}{\|\mathbf{v}\| \|\mathbf{v}'\| (\|\mathbf{v}\| + \|\mathbf{v}'\|)} \right] (\mathbf{\Phi}_k \boldsymbol{\alpha})^\top}_{V} \Phi_k(x). \tag{2.27}$$

In this expression, each Δv_k can be written as a sum $\Delta v_k = \Delta_K v_k + \Delta_S v_k$, where

$$\begin{aligned} \Delta_K v_k &= \mathbf{y}'^\top (\mathbf{K}' + \lambda \mathbf{I})^{-1} \mathbf{K}_k(S') (\mathbf{K}' + \lambda \mathbf{I})^{-1} \mathbf{y}' \\ &\quad - \mathbf{y}'^\top (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{K}_k(S') (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y}' \qquad (2.28) \\ \Delta_S v_k &= \mathbf{y}'^\top (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{K}_k(S') (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y}' \\ &\quad - \mathbf{y}^\top (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{K}_k(S) (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y}. \qquad (2.29) \end{aligned}$$

Let $V = V_1 + V_2$ where V_1 (resp. V_2) is the expression corresponding to Δ_K (resp. Δ_S).

The proof of the propositions giving bounds on $\|V_1\|$ and $\|V_2\|$ are left to the appendix. Our bound on V_2 holds for *orthogonal* base kernels. This assumption is not needed for the bound on $\|V_1\|$, but simplifies the presentation.

Definition 2.2. *Kernels $K_1, \ldots, K_k$ are said to be* orthogonal *if they admit* *feature mappings $\Phi_k\colon \mathcal{X} \mapsto F$ mapping to the same Hilbert space F such that for all $x \in \mathcal{X}$, and $i \neq j$,*

$$\Phi_i(x)^\top \Phi_j(x) = 0. \tag{2.30}$$

This assumption is satisfied in particular by the n-gram based kernels used in our experiments and more generally by kernels K_k whose feature mapping can be obtained by projecting the feature vector $\Phi(x)$ of some kernel K on orthogonal spaces. The *concatenation* type kernels suggested by Bach (2008), are a special case of orthogonal kernels.

Proposition 2.1. *For any samples S and S' differing by one point, the following inequality holds:*

$$\|V_1\| \leq 4\Lambda R\sqrt{pm}\, \|\Delta_K \boldsymbol{\alpha}\|. \tag{2.31}$$

Proposition 2.2. *Assume that the base kernels K_k, $k \in [1, p]$ are orthogonal. Then, for any samples S and S' differing by one point, the following inequality holds:*

$$\|V_2\| \leq \frac{4\Lambda M}{\lambda_{\min} + \lambda_0 m}, \tag{2.32}$$

where we denote by $\lambda_{\min}$ the smaller of $\lambda_{\min}(\mathbf{K})$ and $\lambda_{\min}(\mathbf{K}')$. In order to make the bound on $\|V_1\|$ useful, we further bound $\|\Delta_K \boldsymbol{\alpha}\|$ in terms of $\|V_2\|$ in the following proposition. The proof can be found in the Appendix.

Proposition 2.3. *For any samples S and S' differing by one point, the fol-*

lowing inequality holds:

$$\|\Delta_K \boldsymbol{\alpha}\| \leq \frac{R\sqrt{m}\|V_2\|}{\lambda_{\min} + \lambda_0 m}\,. \tag{2.33}$$

Combining the three propositions leads to

$$\|V\| \leq \frac{4\Lambda M(4\Lambda R^2 p^{1/2}/\lambda_0 + 1)}{\lambda_{\min} + \lambda_0 m}. \tag{2.34}$$

Finally, using the bounds on $|W|$ and $|T|$, which results from the above bound on $\|V\|$ and $\|\Phi_k(x)\| \leq R$, gives the following bound on β_K.

Theorem 2.3. *Assume that the base kernels K_k, $k \in [1, p]$ are orthogonal. Then, for any samples S and S' differing by one point, the following inequality holds:*

$$|\Delta_K h(x)| \leq \frac{4\Lambda M R((4\Lambda R^2 p^{1/2} + R^2)/\lambda_0) + 1)}{\lambda_{\min} + \lambda_0 m} \leq \beta_K\,,$$

where $\lambda_{\min}$ denotes the smaller of $\lambda_{\min}(\mathbf{K})$ and $\lambda_{\min}(\mathbf{K}')$.

Thus, we have $\beta_K \in O(\sqrt{p}/m)$, which is what gives the final bound an additive dependence on the number of base kernels p.

Bound on β

Combining the separate bounds β_S and β_K, gives the final uniform stability bound on β.

Proposition 2.4. *The uniform stability of LKRR, can be bounded as follows:*

$$|\Delta(h(x)-y)^2| \leq 2M|\Delta h(x)| \leq 2M(\beta_S+\beta_K) \leq 2M\frac{C_0+C_1\sqrt{p}}{\lambda_{\min}+\lambda_0 m},$$

with $C_0 = 2R^2M + 4\Lambda RM(R^2/\lambda_0+1)$ *and* $C_1 = 16\Lambda^2 MR^3/\lambda_0$.

A direct application of the general stability bound (Bousquet & Elisseeff, 2002) or the application of McDiarmid's inequality (McDiarmid, 1989) yields the following generalization bound for LKRR.

Theorem 2.4. *Let h denote the hypothesis returned by LKRR and assume that for for all $x \in \mathcal{X}$, $|h(x)-y(x)| \leq M$. Then, for any $\delta > 0$, with probability at least $1-\delta$,*

$$R(h) \leq \widehat{R}(h) + 2\beta + \big(4m\beta + M\big)\sqrt{\frac{\log\frac{1}{\delta}}{2m}},$$

where $\beta = 2M(\beta_S+\beta_K)$, with $\beta_S = O(1/m)$ and $\beta_K = O(\sqrt{p}/m)$, is the stability bound given by Proposition 2.4.

Thus, in view of this theorem, our generalization bound has only an additional additive term that depends on the number of kernels, which is of the form $O(\sqrt{p/m})$. Note that in the case of a fixed kernel, all Δ_K terms are zero and thus $\beta_K = 0$. In such a case $\beta = 2M\beta_S = O(1/m)$ and results in a generalization bound of the form $R(h) \leq \widehat{R}(h) + O(1/\sqrt{m})$, which matches exactly the fixed kernel bound given by Bousquet and Elisseeff (2002) and is in fact tighter in terms of constants.

Most importantly, this bound ensures that is it reasonable to use a relatively large number of kernels in this regression setting without over-fitting. The results of Section 3.2 will corroborate these theoretical findings with empirical results.

2.4 Rademacher-Based Proofs

In this section, we present new generalization bounds for the family of convex combinations of base kernels and an L_1 constraint that have only a logarithmic dependency on p. Our learning bounds are based on a careful analysis of the Rademacher complexity of the hypothesis set considered and has the form: $R(h) \leq \widehat{R}_\rho(h) + O\Big(\sqrt{\frac{(\log p)R^2/\rho^2}{m}}\Big)$. Our bound is simpler and contains no other extra logarithmic term. Thus, this represents a substantial improvement over the previous best bounds for this problem. Our bound is also valid for a very large number of kernels, in particular for $p \gg m$, while the previous bounds were not informative in that case.

We also present new generalization bounds for the family of non-negative combinations of base kernels with an L_2 regularization and L_q regularization with other values of $q > 1$. An algorithm specific stability bound was given in Section 3.2 that had only an additive dependency with respect to p, assuming a technical condition of orthogonality on the base kernels. The learning bound for L_2 regularization presented in this section does not require any assumption on the family of base kernels. It admits only a mild multiplicative dependency of $p^{1/4}$ on the number of base kernels.

Most learning kernel algorithms are based on a hypothesis set derived from convex combinations of a fixed set of $p \geq 1$ kernels $K_1, \ldots, K_p$:

$$H_p^1 = \Big\{x \mapsto \mathbf{w} \cdot \mathbf{\Phi}_K(x) \colon K = \sum_{k=1}^{p} \mu_k K_k, \mu_k \geq 0, \sum_{k=1}^{p} \mu_k = 1, \|\mathbf{w}\| \leq 1\Big\}.$$

We consider more generally the hypothesis sets H_p^q, $q \geq 1$, based on a L_q constraint on the vector $\boldsymbol{\mu}$ and defined as follows:

$$H_p^q = \Big\{x \mapsto \mathbf{w} \cdot \mathbf{\Phi}_K(x) \colon K = \sum_{k=1}^{p} \mu_k K_k, \mu_k \geq 0, \sum_{k=1}^{p} \mu_k^q = 1, \|\mathbf{w}\| \leq 1\Big\}.$$

We bound, for different values of q, including $q = 1$ and $q = 2$, the empirical Rademacher complexity $\widehat{\mathfrak{R}}_S(H_p^q)$ of these families for an arbitrary sample S of size m, which immediately yields a generalization bound for learning kernels based on these families of hypotheses. For a fixed sample $S = (x_1, \ldots, x_m)$, the empirical Rademacher complexity of a hypothesis set H is defined as

$$\widehat{\mathfrak{R}}_S(H) = \frac{1}{m} \operatorname*{E}_{\boldsymbol{\sigma}} \Big[\sup_{h \in H} \sum_{i=1}^{m} \sigma_i h(x_i)\Big],$$

where the expectation is taken over $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_m)^\top$ where $\sigma_i \in \{-1, +1\}$, $i \in [1, m]$, are independent uniform random variables.

Let $\mathbf{w}_S = \sum_{i=1}^{m} \alpha_i \mathbf{\Phi}_K(x_i)$ be the orthogonal projection of $\mathbf{w}$ on $\mathbb{H}_S = \operatorname{span}(\mathbf{\Phi}_K(x_1), \ldots, \mathbf{\Phi}_K(x_m))$. Then, $\mathbf{w}$ can be written as $\mathbf{w} = \mathbf{w}_S + \mathbf{w}^\perp$, with $\mathbf{w}_S \cdot \mathbf{w}^\perp = 0$. Thus, $\|\mathbf{w}\|^2 = \|\mathbf{w}_S\|^2 + \|\mathbf{w}^\perp\|^2$, which, in view of $\|\mathbf{w}\| \leq 1$

implies $\|\mathbf{w}_S\|^2 \leq 1$. Since $\|\mathbf{w}_S\|^2 = \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$, this implies

$$\boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \leq 1. \tag{2.35}$$

Observe also that for all $x \in S$,

$$h(x) = \mathbf{w} \cdot \mathbf{\Phi}_K(x) = \mathbf{w}_S \cdot \mathbf{\Phi}_K(x) = \sum_{i=1}^{m} \alpha_i K(x_i, x). \tag{2.36}$$

Conversely, any function $\sum_{i=1}^{m} \alpha_i K(x_i, \cdot)$ with $\boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \leq 1$ is clearly an element of H_p^1.

Proposition 2.5. *Let $q, r \geq 1$ with $\frac{1}{q} + \frac{1}{r} = 1$. For any sample S of size m, the empirical Rademacher complexity of the hypothesis set H_p^q can be expressed as*

$$\widehat{\mathfrak{R}}_S(H_p^q) = \frac{1}{m} \operatorname*{E}_{\boldsymbol{\sigma}} \left[\sqrt{\|\mathbf{u}_{\boldsymbol{\sigma}}\|_r} \right]$$

with $\mathbf{u}_{\boldsymbol{\sigma}} = (\boldsymbol{\sigma}^\top \mathbf{K}_1 \boldsymbol{\sigma}, \ldots, \boldsymbol{\sigma}^\top \mathbf{K}_p \boldsymbol{\sigma})^\top$.

Proof. Fix a sample $S = (x_1, \ldots, x_m)$, and denote by $\mathcal{M}_q = \{\boldsymbol{\mu} \geq 0 \colon \|\boldsymbol{\mu}\|_q = 1\}$ and by $\mathcal{A} = \{\boldsymbol{\alpha} \colon \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \leq 1\}$. Then, in view of (2.35) and (2.36), the

Rademacher complexity $\widehat{\mathfrak{R}}_S(H_p^q)$ can be expressed as follows:

$$\begin{aligned}\widehat{\mathfrak{R}}_S(H_p^q) &= \frac{1}{m} \operatorname*{E}_{\boldsymbol{\sigma}} \Big[\sup_{h \in H_p^q} \sum_{i=1}^m \sigma_i h(x_i) \Big] \\ &= \frac{1}{m} \operatorname*{E}_{\boldsymbol{\sigma}} \Big[\sup_{\boldsymbol{\mu} \in \mathcal{M}_q, \boldsymbol{\alpha} \in \mathcal{A}} \sum_{i,j=1}^m \sigma_i \alpha_j K(x_i, x_j) \Big] \\ &= \frac{1}{m} \operatorname*{E}_{\boldsymbol{\sigma}} \Big[\sup_{\boldsymbol{\mu} \in \mathcal{M}_q, \boldsymbol{\alpha} \in \mathcal{A}} \boldsymbol{\sigma}^\top \mathbf{K} \boldsymbol{\alpha} \Big].\end{aligned}$$

Now, by the Cauchy-Schwarz inequality, the supremum $\sup_{\boldsymbol{\alpha} \in \mathcal{A}} \boldsymbol{\sigma}^\top \mathbf{K} \boldsymbol{\alpha}$ is reached for $\boldsymbol{\alpha}$ collinear with $\boldsymbol{\sigma}$, which gives $\sup_{\boldsymbol{\alpha} \in \mathcal{A}} \boldsymbol{\sigma}^\top \mathbf{K} \boldsymbol{\alpha} = \sqrt{\boldsymbol{\sigma}^\top \mathbf{K} \boldsymbol{\sigma}}$. Thus,

$$\widehat{\mathfrak{R}}_S(H_p^q) = \frac{1}{m} \operatorname*{E}_{\boldsymbol{\sigma}} \Big[\sup_{\boldsymbol{\mu} \in \mathcal{M}_q} \sqrt{\boldsymbol{\sigma}^\top \mathbf{K} \boldsymbol{\sigma}} \Big] = \frac{1}{m} \operatorname*{E}_{\boldsymbol{\sigma}} \Big[\sup_{\boldsymbol{\mu} \in \mathcal{M}_q} \sqrt{\boldsymbol{\mu} \cdot \mathbf{u}_{\boldsymbol{\sigma}}} \Big].$$

By the definition of the dual norm, $\sup_{\boldsymbol{\mu} \in \mathcal{M}_q} \boldsymbol{\mu} \cdot \mathbf{u}_{\boldsymbol{\sigma}} = \|\mathbf{u}_{\boldsymbol{\sigma}}\|_r$, which gives $\widehat{\mathfrak{R}}_S(H_p^q) = \frac{1}{m} \mathrm{E}_{\boldsymbol{\sigma}} \big[\sqrt{\|\mathbf{u}_{\boldsymbol{\sigma}}\|_r} \big]$. □

2.4.1 Rademacher Complexity Bound for H_p^1

Our bounds on the empirical Rademacher complexity of the families H_p^1 or H_p^q for $q = 2$ or other values of q relies on the following result.

Lemma 2.1. *Let $\mathbf{K}$ be the kernel matrix of a kernel function K associated to a sample S. Then, for any integer r, the following inequality holds:*

$$\operatorname*{E}_{\boldsymbol{\sigma}} \Big[(\boldsymbol{\sigma}^\top \mathbf{K} \boldsymbol{\sigma})^r \Big] \leq \Big(\eta_0 r \operatorname{Tr}[\mathbf{K}] \Big)^r,$$

where $\eta_0 = \frac{23}{22}$.

Proof. We use a combinatorial argument to bound the expectation. Since r is an integer, we can write:

$$\begin{aligned}
&\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\left[(\boldsymbol{\sigma}^\top \mathbf{K}\boldsymbol{\sigma})^r\right] = \mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\left[\Big(\sum_{i,j=1}^m \sigma_i\sigma_j K(x_i,x_j)\Big)^r\right]\\
&= \sum_{\substack{1\le i_1,\ldots,i_r\le m\\ 1\le j_1,\ldots,j_r\le m}} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\left[\prod_{s=1}^r \sigma_{i_s}\sigma_{j_s}\right]\prod_{s=1}^r K(x_{i_s},x_{j_s})\\
&\le \sum_{\substack{1\le i_1,\ldots,i_r\le m\\ 1\le j_1,\ldots,j_r\le m}} \left|\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\left[\prod_{s=1}^r \sigma_{i_s}\sigma_{j_s}\right]\right|\prod_{s=1}^r |K(x_{i_s},x_{j_s})|\\
&\le \sum_{\substack{1\le i_1,\ldots,i_r\le m\\ 1\le j_1,\ldots,j_r\le m}} \left|\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\left[\prod_{s=1}^r \sigma_{i_s}\sigma_{j_s}\right]\right|\\
&\qquad \prod_{s=1}^r \sqrt{K(x_{i_s},x_{i_s})K(x_{j_s},x_{j_s})} \quad \text{(Cauchy-Schwarz)}\\
&= \sum_{s_1+\ldots+s_m=2r} \binom{2r}{s_1,\ldots,s_m}\left|\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}[\sigma_1^{s_1}\cdots\sigma_m^{s_m}]\right|\\
&\qquad\qquad \sqrt{K(x_1,x_1)^{s_1}\cdots K(x_m,x_m)^{s_m}}.
\end{aligned}$$

Since $\mathrm{E}[\sigma_i]=0$ for all i and since the Rademacher variables are independent, we can write $\mathrm{E}[\sigma_{i_1}\ldots\sigma_{i_l}] = \mathrm{E}[\sigma_{i_1}]\cdots\mathrm{E}[\sigma_{i_l}] = 0$ for any l distinct variables $\sigma_{i_1},\ldots,\sigma_{i_l}$. Thus, $\mathrm{E}_{\boldsymbol{\sigma}}\left[\sigma_1^{s_1}\cdots\sigma_1^{s_m}\right] = 0$ unless all s_is are even, in which case $\mathrm{E}_{\boldsymbol{\sigma}}\left[\sigma_1^{s_1}\cdots\sigma_m^{s_m}\right] = 1$. It follows that:

$$\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\left[(\boldsymbol{\sigma}^\top \mathbf{K}\boldsymbol{\sigma})^r\right] \le \sum_{2t_1+\ldots+2t_m=2r}\binom{2r}{2t_1,\ldots,2t_m}\prod_{i=1}^m K(x_i,x_i)^{t_i}.$$

By Lemma A.2 (see Appendix, Section A.2), each multinomial coefficient $\binom{2r}{2t_1,\ldots,2t_m}$ can be bounded by $(\eta_0 r)^r \binom{r}{t_1,\ldots,t_m}$, where $\eta_0 = 1 + \frac{1}{22}$. This gives

$$\begin{aligned}
\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\Big[(\boldsymbol{\sigma}^\top \mathbf{K}\boldsymbol{\sigma})^r\Big] &\leq (\eta_0 r)^r \sum_{t_1+\ldots+t_m=r} \binom{r}{t_1,\ldots,t_m} \prod_{i=1}^m K(x_i,x_i)^{t_i} \\
&= (\eta_0 r)^r (\mathrm{Tr}[\mathbf{K}])^r = \Big(\eta_0 r\,\mathrm{Tr}[\mathbf{K}]\Big)^r,
\end{aligned}$$

which coincides with the statement of the lemma. □

Theorem 2.5. *For any sample S of size m, the empirical Rademacher complexity of the hypothesis set H_p^1 can be bounded as follows:*

$$\forall r \in \mathbb{N}, r \geq 1, \quad \widehat{\mathfrak{R}}_S(H_p^1) \leq \frac{\sqrt{\eta_0 r \|\mathbf{u}\|_r}}{m},$$

where $\mathbf{u} = (\mathrm{Tr}[\mathbf{K}_1], \ldots, \mathrm{Tr}[\mathbf{K}_p])^\top$ and $\eta_0 = \frac{23}{22}$.

Proof. By Proposition 2.5, $\widehat{\mathfrak{R}}_S(H_p^1) = \frac{1}{m}\mathrm{E}_{\boldsymbol{\sigma}}\big[\sqrt{\|\mathbf{u}_{\boldsymbol{\sigma}}\|_\infty}\big]$. Since for any $r \geq 1$, $\|\mathbf{u}_{\boldsymbol{\sigma}}\|_\infty \leq \|\mathbf{u}_{\boldsymbol{\sigma}}\|_r$, we can upper bound the Rademacher complexity as follows:

$$\begin{aligned}
\widehat{\mathfrak{R}}_S(H_p^1) &\leq \frac{1}{m}\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\big[\sqrt{\|\mathbf{u}_{\boldsymbol{\sigma}}\|_r}\big] \\
&= \frac{1}{m}\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\Big[\Big[\sum_{k=1}^p (\boldsymbol{\sigma}^\top \mathbf{K}_k \boldsymbol{\sigma})^r\Big]^{\frac{1}{2r}}\Big] \\
&\leq \frac{1}{m}\Big[\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\Big[\sum_{k=1}^p (\boldsymbol{\sigma}^\top \mathbf{K}_k \boldsymbol{\sigma})^r\Big]\Big]^{\frac{1}{2r}} \text{ (Jensen's inequality)} \\
&= \frac{1}{m}\Big[\sum_{k=1}^p \mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\Big[(\boldsymbol{\sigma}^\top \mathbf{K}_k \boldsymbol{\sigma})^r\Big]\Big]^{\frac{1}{2r}}.
\end{aligned}$$

Assume that $r \geq 1$ is an integer, then, by Lemma 2.1, for any $k \in [1, p]$, we have

$$\operatorname*{E}_{\boldsymbol{\sigma}}\Big[(\boldsymbol{\sigma}^\top \mathbf{K}_k \boldsymbol{\sigma})^r\Big] \leq \Big(\eta_0 r \operatorname{Tr}[\mathbf{K}_k]\Big)^r.$$

Using these inequalities gives

$$\widehat{\mathfrak{R}}_S(H_p^1) \leq \frac{1}{m}\Big[\sum_{k=1}^{p}\Big(\eta_0 r \operatorname{Tr}[\mathbf{K}_k]\Big)^r\Big]^{\frac{1}{2r}} = \frac{\sqrt{\eta_0 r \|\mathbf{u}\|_r}}{m},$$

and concludes the proof. □

Theorem 2.6. *Let $p > 1$ and assume that $K_k(x, x) \leq R^2$ for all $x \in \mathcal{X}$ and $k \in [1, p]$. Then, for any sample S of size m, the Rademacher complexity of the hypothesis set H_p^1 can be bounded as follows:*

$$\widehat{\mathfrak{R}}_S(H_p^1) \leq \sqrt{\frac{\eta_0 e \lceil \log p \rceil R^2}{m}}.$$

Proof. Since $K_k(x, x) \leq R^2$ for all $x \in \mathcal{X}$ and $k \in [1, p]$, $\operatorname{Tr}[\mathbf{K}_k] \leq mR^2$ for all $k \in [1, p]$. Thus, by Theorem 2.5, for any integer $r > 1$, the Rademacher complexity can be bounded as follows

$$\widehat{\mathfrak{R}}_S(H_p^1) \leq \frac{1}{m}\Big[p\Big(\eta_0 r m R^2\Big)^r\Big]^{\frac{1}{2r}} = \sqrt{\frac{\eta_0 r p^{\frac{1}{r}} R^2}{m}}.$$

For $p > 1$, the function $r \mapsto p^{1/r} r$ reaches its minimum at $r_0 = \log p$, which gives $\widehat{\mathfrak{R}}_S(H_p^1) \leq \sqrt{\frac{\eta_0 e \lceil \log p \rceil R^2}{m}}$. □

Note that more generally, without assuming $K_k(x, x) \leq R^2$ for all k and all x, the same proof yields the following result:

$$\widehat{\mathfrak{R}}_S(H_p^1) \leq \sqrt{\frac{\eta_0 e \lceil \log p \rceil \|\mathbf{u}\|_\infty}{m}}.$$

Remarkably, the bound of the theorem has a very mild dependence on p. The theorem can be used to derive generalization bounds for learning kernels in classification, regression, and other tasks. We briefly illustrate its application to binary classification where the labels y are in $\{-1, +1\}$. Let $R(h)$ denote the generalization error of $h \in H_p^1$, that is $R(h) = \Pr[yh(x) < 0]$. For a training sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$ and any $\rho > 0$, define the ρ-empirical margin loss $\widehat{R}_\rho(h)$ as follows:

$$\widehat{R}_\rho(h) = \frac{1}{m} \sum_{i=1}^{m} \min\left(1, [1 - y_i h(x_i)/\rho]_+\right).$$

Note that $\widehat{R}_\rho(h)$ is always upper bounded by the fraction of the training points with margin less than ρ:

$$\widehat{R}_\rho(h) \leq \frac{1}{m} \sum_{i=1}^{m} 1_{y_i h(x_i) < \rho}.$$

The following gives a margin-based generalization bound for the hypothesis set H_p^1.

Corollary 2.1. *Fix $\rho > 0$. Then, for any integer $r > 1$, for any $\delta > 0$, with* *probability at least* $1-\delta$*, for any* $h \in H_p^1$*,*

$$R(h) \leq \widehat{R}_\rho(h) + \frac{2\sqrt{\eta_0 r \|\mathbf{u}\|_r}}{m\rho} + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$

with $\mathbf{u} = (\mathrm{Tr}[\mathbf{K}_1], \ldots, \mathrm{Tr}[\mathbf{K}_p])^\top$ *and* $\eta_0 = \frac{23}{22}$.

If additionally, $K_k(x,x) \leq R^2$ *for all* $x \in \mathcal{X}$ *and* $k \in [1,p]$*, then, for* $p > 1$*,*

$$R(h) \leq \widehat{R}_\rho(h) + 2\sqrt{\frac{\eta_0 e \lceil \log p \rceil R^2/\rho^2}{m}} + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$

Proof. With our definition of the Rademacher complexity, for any $\delta > 0$, with probability at least $1-\delta$, the following bound holds for any $h \in H_p^1$ (Koltchinskii & Panchenko, 2002; Bartlett & Mendelson, 2002):

$$R(h) \leq \widehat{R}_\rho(h) + \frac{2}{\rho}\widehat{\mathfrak{R}}_S(H_p^1) + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$

Plugging in the bound on the empirical Rademacher complexity given by Theorem 2.5 and Theorem 2.6 yields the statement of the corollary. □

The bound of the Corollary can be straightforwardly extended to hold uniformly over all choices of ρ, using standard techniques introduced by Koltchinskii and Panchenko (2002), at the price of the additional term $\frac{\log\log_2(4R/\rho)}{m}$ on the right-hand side.

2.4.2 Rademacher Complexity Bound for H_p^q

This section presents bounds on the Rademacher complexity of the hypothesis sets H_p^q for various values of $q > 1$, including $q = 2$.

Theorem 2.7. *Let $q, r \geq 1$ with $\frac{1}{q} + \frac{1}{r} = 1$ and assume that r is an integer. Then, for any sample S of size m, the empirical Rademacher complexity of the hypothesis set H_p^q can be bounded as follows:*

$$\widehat{\mathfrak{R}}_S(H_p^q) \leq \frac{\sqrt{\eta_0 r \|\mathbf{u}\|_r}}{m},$$

where $\mathbf{u} = (\mathrm{Tr}[\mathbf{K}_1], \ldots, \mathrm{Tr}[\mathbf{K}_p])^\top$ and $\eta_0 = \frac{23}{22}$.

Proof. By Proposition 2.5, $\widehat{\mathfrak{R}}_S(H_p^q) = \frac{1}{m} \mathrm{E}_{\boldsymbol{\sigma}} \left[\sqrt{\|\mathbf{u}_{\boldsymbol{\sigma}}\|_r}\right]$, with $\mathbf{u}_{\boldsymbol{\sigma}} = (\boldsymbol{\sigma}^\top \mathbf{K}_1 \boldsymbol{\sigma}, \ldots, \boldsymbol{\sigma}^\top \mathbf{K}_p \boldsymbol{\sigma})^\top$. The rest of the proof is identical to that of Theorem 2.5: using Jensen's inequality and Lemma 2.1, which applies because r is an integer, we obtain similarly

$$\widehat{\mathfrak{R}}_S(H_p^q) \leq \frac{1}{m} \Big[\sum_{k=1}^p \Big(\eta_0 r \, \mathrm{Tr}[\mathbf{K}_k] \Big)^r \Big]^{\frac{1}{2r}}. \qquad \square$$

In particular, for $q = r = 2$, the theorem implies

$$\widehat{\mathfrak{R}}_S(H_p^2) \leq \frac{\sqrt{2\eta_0 \|\mathbf{u}\|_2}}{m}.$$

Theorem 2.8. *Let $q, r \geq 1$ with $\frac{1}{q} + \frac{1}{r} = 1$ and assume that r is an integer. Let $p > 1$ and assume that $K_k(x, x) \leq R^2$ for all $x \in \mathcal{X}$ and $k \in [1, p]$. Then,* *for any sample S of size m, the Rademacher complexity of the hypothesis set H_p^q can be bounded as follows:*

$$\widehat{\mathfrak{R}}_S(H_p^q) \leq \sqrt{\frac{\eta_0 r p^{\frac{1}{r}} R^2}{m}}.$$

Proof. Since $K_k(x, x) \leq R^2$ for all $x \in \mathcal{X}$ and $k \in [1, p]$, $\mathrm{Tr}[\mathbf{K}_k] \leq mR^2$ for all $k \in [1, p]$. Thus, by Theorem 2.7, the Rademacher complexity can be bounded as follows

$$\widehat{\mathfrak{R}}_S(H_p^q) \leq \frac{1}{m}\left[p\Big(\eta_0 r m R^2\Big)^r\right]^{\frac{1}{2r}} = \sqrt{\frac{\eta_0 r p^{\frac{1}{r}} R^2}{m}}. \qquad \square$$

The bound of the theorem has only a mild dependence ($\sqrt[2r]{\cdot}$) on the number of kernels p. In particular, for $q = r = 2$, under the assumptions of the theorem,

$$\widehat{\mathfrak{R}}_S(H_p^2) \leq \sqrt{\frac{2\eta_0 \sqrt{p} R^2}{m}},$$

and the dependence is in $O(p^{1/4})$.

Proceeding as in the L_1 case leads to the following margin bound in binary classification.

Corollary 2.2. *Let $q, r \geq 1$ with $\frac{1}{q} + \frac{1}{r} = 1$ and assume that r is an integer. Fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, for any $h \in H_p^q$,*

$$R(h) \leq \widehat{R}_\rho(h) + \frac{2\sqrt{\eta_0 r \|\mathbf{u}\|_r}}{m\rho} + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$

with $\mathbf{u} = (\mathrm{Tr}[\mathbf{K}_1], \ldots, \mathrm{Tr}[\mathbf{K}_p])^\top$ and $\eta_0 = \frac{23}{22}$.

If additionally, $K_k(x, x) \leq R^2$ for all $x \in \mathcal{X}$ and $k \in [1, p]$, then, for $p > 1$,

$$R(h) \leq \widehat{R}_\rho(h) + 2\sqrt{\frac{\eta_0 r p^{\frac{1}{r}} R^2/\rho^2}{m}} + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$

In particular, for $q = r = 2$, the generalization bound of the corollary becomes

$$R(h) \leq \widehat{R}_\rho(h) + 2\sqrt{\frac{\eta_0 r \sqrt{p} R^2/\rho^2}{m}} + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$

The $p^{1/(2r)}$ dependency of the generalization bound of Corollary 2.2 cannot be improved. In particular, the $p^{1/4}$ dependency is tight for the the hypothesis set H_p^2. Indeed, as shown by Koltchinskii and Panchenko (2002) using the family of canonical projections $\Phi \colon x \mapsto x_k$, in general, the term $\mathfrak{R}_m(H)/\rho$ cannot be improved upon for such margin-based generalization bounds for a hypothesis set H. By Proposition 2.5, $\widehat{\mathfrak{R}}_m(H_p^q) = p^{1/(2r)} \frac{1}{m} \operatorname{E}\left[\sqrt{\sigma^\top \mathbf{K}_1 \sigma}\right]$ when all kernels K_k are equal. Thus, this shows that in general the dependency on $p^{1/(2r)}$ is necessary.

2.4.3 Proof Techniques

Our proof techniques are somewhat general and apply similarly to other problems. In particular, they can be used as alternative methods to derive bounds on the Rademacher complexity of linear functions classes, such as those given by Kakade et al. (2009), using strong convexity. In fact, in some cases, they can lead to similar bounds but with tighter constants. The following theorem illustrates that in the case of linear functions constrained by the norm $\|\cdot\|_q$.

Theorem 2.9. *Let $q, r \geq 1$ with $\frac{1}{q} + \frac{1}{r} = 1$, r an even integer such that $r \geq 2$. Let $\mathcal{X} = \{x \colon \|x\|_r \leq X\}$, and let $\mathcal{F}$ be the class of linear functions over $\mathcal{X}$ defined by $\mathcal{F} = \{\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} \colon \|\mathbf{w}\|_q \leq W\}$, then, for any sample $S = (x_1, \ldots, x_m)$, the following bound holds for the empirical Rademacher complexity of this class:*

$$\widehat{\mathfrak{R}}_S(\mathcal{F}) \leq XW\sqrt{\frac{\eta_0 r}{2m}}.$$

Clearly, this immediately yields the same bound on the Rademacher complexity $\mathfrak{R}_m(\mathcal{F}) = \mathrm{E}_S[\widehat{\mathfrak{R}}_S(\mathcal{F})]$. The bound given by Kakade et al. (2009)[Section 3.1] in this case is $\mathfrak{R}_m(\mathcal{F}) \leq XW\sqrt{\frac{r-1}{m}}$. Since $\eta_0 r/2 \leq r - 1$, for an even integer $r > 2$, our bound is always tighter.

Proof. The proof is similar to and uses that of Theorem 2.5. By the definition of the dual norms, the following holds:

$$\widehat{\mathfrak{R}}_S(\mathcal{F}) = \frac{1}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\Big[\sup_{\|\mathbf{w}\|_q \leq W} \sum_{i=1}^m \sigma_i \mathbf{w} \cdot \mathbf{x}_i\Big] = \frac{W}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\Big[\Big\|\sum_{i=1}^m \sigma_i \mathbf{x}_i\Big\|_r\Big].$$

By Jensen's inequality,

$$\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\Big\|\sum_{i=1}^m \sigma_i \mathbf{x}_i\Big\|_r \leq \Big[\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\Big\|\sum_{i=1}^m \sigma_i \mathbf{x}_i\Big\|_r^r\Big]^{\frac{1}{r}} = \Big[\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\sum_{j=1}^N \big[\sum_{i=1}^m \sigma_i x_{ij}\big]^r\Big]^{\frac{1}{r}},$$

where we denote by N the dimension of the space and by x_{ij} the jth coordinate of $\mathbf{x}_i$. Now, we can bound the term $\mathrm{E}_{\boldsymbol{\sigma}}\Big[\big[\sum_{i=1}^m \sigma_i x_{ij}\big]^r\Big]$ using Lemma 2.1 and obtain:

$$\begin{aligned}\mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\Big[\big[\sum_{i=1}^m \sigma_i x_{ij}\big]^r\Big] &= \mathop{\mathrm{E}}_{\boldsymbol{\sigma}}\Big[\big[\sum_{i,l=1}^m \sigma_i \sigma_l x_{ij} x_{lj}\big]^{r/2}\Big] \\ &\leq \Big(\frac{\eta_0 r}{2}\sum_{i=1}^m x_{ij}^2\Big)^{r/2}.\end{aligned}$$

Thus,

$$\begin{aligned}\widehat{\mathfrak{R}}_S(\mathcal{F}) &\leq \frac{W}{m}\Big(\frac{\eta_0 r}{2}\Big)^{1/2}\Big[\sum_{j=1}^N \Big(\sum_{i=1}^m x_{ij}^2\Big)^{r/2}\Big]^{\frac{1}{r}} \\ &= W\sqrt{\frac{\eta_0 r}{2m}}\Big[\sum_{j=1}^N \Big(\frac{1}{m}\sum_{i=1}^m x_{ij}^2\Big)^{r/2}\Big]^{\frac{1}{r}}.\end{aligned}$$

Since $r \geq 2$, by Jensen's inequality, $\Big(\frac{1}{m}\sum_{i=1}^m x_{ij}^2\Big)^{r/2} \leq \frac{1}{m}\sum_{i=1}^m x_{ij}^r$. Thus,

$$\begin{aligned}\widehat{\mathfrak{R}}_S(\mathcal{F}) &\leq W\sqrt{\frac{\eta_0 r}{2m}}\Big[\sum_{j=1}^N \frac{1}{m}\sum_{i=1}^m x_{ij}^r\Big]^{\frac{1}{r}} \\ &= W\sqrt{\frac{\eta_0 r}{2m}}\Big[\frac{1}{m}\sum_{i=1}^m \|\mathbf{x}_i\|_r^r\Big]^{\frac{1}{r}} \leq W\sqrt{\frac{\eta_0 r}{2m}}X. \quad \square\end{aligned}$$

In this section we have presented several new generalization bounds for the problem of learning kernels with non-negative combinations of base kernels and outlined the relevance of our proof techniques to the analysis of the complexity of the class of linear functions. The bounds are simpler and significantly improve over previous bounds. Their behavior matches empirical observations with a large number of base kernels. Their very mild dependency on the number of kernels suggests the use of a *very* large number of kernels is possible for this problem. Recent experiments by Bach (2008) in regression, as well as experiments shown in Chapter 3, using a large number of kernels seem to corroborate this idea.

2.5 Theoretical Results for Alignment

In Section 3.3 we explore the empirical performance of a *two-stage* technique and algorithm for learning kernels. The first stage of this technique consists of *learning* a kernel K that is a convex combination of p kernels. The second stage consists of using K with a standard kernel-based learning algorithm such as support vector machines (SVMs) (Cortes & Vapnik, 1995) for classification, or KRR (Saunders et al., 1998) for regression, to select a prediction hypothesis.

The main motivation for using a two-stage method, is that there are fewer parameters to learn simultaneously and better performance may be observed when data is scarce. Here, we first learn a kernel with p parameters and then using the selected kernel learn a hypothesis with m parameters. In the previously suggested one-stage methods both set of parameters are learned jointly, resulting in a more complicated model that requires possibly more data in order to generalize.

Different methods can be used to learn, from the training sample, the convex combination of parameters that define K. A measure of similarity between the base kernels K_k, $k \in [1, p]$, and the target kernel K_Y derived from the labels can be used to determine these parameters. This can be done by using either the individual similarity of each kernel K_k with K_Y, or globally, from the similarity between convex combinations of the base kernels and K_Y. The similarities we consider are based on the natural notion of *kernel alignment* introduced by Cristianini et al. (2001), though our definition differs from the original one. We note that other measures of similarity could be used in this context. In particular, the notion of similarity suggested by Balcan and Blum (2006) could be used if it was computable from finite samples.

In this section we present a number of novel theoretical results for the alignment-based two-stage techniques. Our results build on previous work by Cristianini et al. (2001); Cristianini et al. (2002); Kandola et al. (2002a), but we significantly extend that work in several directions. We discuss the original definitions of kernel alignment by these authors and adopt a related but different definition. We give a novel concentration bound showing that the difference between the alignment of two kernel matrices and the alignment of the corresponding kernel functions can be bounded by a term in $O(1/\sqrt{m})$. Our result is simpler and directly bounds the difference between the relevant quantities, unlike previous work. We also show the existence of good predictors for kernels with high alignment, both for classification and for regression. These results correct a technical problem in classification and extend to regression the bounds of Cristianini et al. (2001).

2.5.1 Alignment Definitions

The notion of kernel alignment was first introduced by Cristianini et al. (2001). Our definition of kernel alignment is different and is based on the notion of centering in the feature space. Thus, we start with the definition of centering and the analysis of its relevant properties.

Centering Kernels

Let D be the distribution according to which training and test points are drawn. Centering a feature mapping $\Phi\colon \mathcal{X} \to H$ consists of replacing it by $\Phi - \mathrm{E}_x[\Phi]$, where E_x denotes the expected value of Φ when x is drawn according to the distribution D. Centering a positive semi-definite, PSD, kernel function $K\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ consists of centering any feature mapping Φ associated to K. Thus, the centered kernel K_c associated to K is defined for all $x, x' \in \mathcal{X}$ by

$$\begin{aligned} K_c(x, x') &= (\Phi(x) - \mathop{\mathrm{E}}_{x}[\Phi])^\top (\Phi(x') - \mathop{\mathrm{E}}_{x'}[\Phi]) & (2.37) \\ &= K(x, x') - \mathop{\mathrm{E}}_{x}[K(x, x')] - \mathop{\mathrm{E}}_{x'}[K(x, x')] + \mathop{\mathrm{E}}_{x,x'}[K(x, x')]. & (2.38) \end{aligned}$$

This also shows that the definition does not depend on the choice of the feature mapping associated to K. Since $K_c(x, x')$ is defined as an inner product, K_c is also a PSD kernel. Note also that for a centered kernel K_c, $\mathrm{E}_{x,x'}[K_c(x, x')] = 0$. That is, centering the feature mapping implies centering the kernel function.

Similar definitions can be given for a finite sample $S = (x_1, \ldots, x_m)$ drawn according to D: a feature vector $\Phi(x_i)$ with $i \in [1, m]$ is then centered by replacing it with $\Phi(x_i) - \overline{\Phi}$, with $\overline{\Phi} = \frac{1}{m}\sum_{i=1}^m \Phi(x_i)$, and the kernel matrix $\mathbf{K}$ associated to K and the sample S is centered by replacing it with $\mathbf{K}_c$ defined for all $i, j \in [1, m]$ by

$$[\mathbf{K}_c]_{ij} = \mathbf{K}_{ij} - \frac{1}{m}\sum_{i=1}^m \mathbf{K}_{ij} - \frac{1}{m}\sum_{j=1}^m \mathbf{K}_{ij} + \frac{1}{m^2}\sum_{i,j=1}^m \mathbf{K}_{ij}. \tag{2.39}$$

Let $\mathbf{\Phi} = [\Phi(x_1), \ldots, \Phi(x_m)]^\top$ and $\overline{\mathbf{\Phi}} = [\overline{\Phi}, \ldots, \overline{\Phi}]^\top$. Then, it is not hard to verify that $\mathbf{K}_c = (\mathbf{\Phi} - \overline{\mathbf{\Phi}})(\mathbf{\Phi} - \overline{\mathbf{\Phi}})^\top$, which shows that $\mathbf{K}_c$ is a positive semi-definite matrix. Also, as with the kernel function, $\frac{1}{m^2}\sum_{i,j=1}^m [\mathbf{K}_c]_{ij} = 0$.

Kernel Alignment

We define the alignment of two kernel functions as follows.

Definition 2.3. *Let K and K' be two kernel functions defined over $\mathcal{X} \times \mathcal{X}$ such that $0 < \mathrm{E}[K_c^2] < +\infty$ and $0 < \mathrm{E}[{K'_c}^2] < +\infty$. Then, the* alignment *between K and K' is defined by*

$$\rho(K, K') = \frac{\mathrm{E}[K_c K'_c]}{\sqrt{\mathrm{E}[K_c^2]\,\mathrm{E}[{K'_c}^2]}} \ .$$

In the absence of ambiguity, to abbreviate the notation, we often omit the variables over which an expectation is taken. Since $|\,\mathrm{E}[K_c K'_c]| \leq \sqrt{\mathrm{E}[K_c^2]\,\mathrm{E}[{K'_c}^2]}$ by the Cauchy-Shwarz inequality, we have $\rho(K, K') \in [-1, 1]$. The following lemma shows more precisely that $\rho(K, K') \in [0, 1]$ when K_c and K'_c are PSD kernels. We denote by $\langle \cdot, \cdot \rangle_F$ the Frobenius product and by $\| \cdot \|_F$ the Frobenius norm.

Lemma 2.2. *For any two PSD kernels Q and Q', $\mathrm{E}[QQ'] \geq 0$.*

Proof. Let Ψ be a feature mapping associated to Q and Ψ' a feature mapping associated to Q'. By definition of Ψ and Ψ', and using the properties of the trace, we can write:

$$\begin{aligned}
\mathop{\mathrm{E}}_{x,x'}[Q(x,x')Q'(x,x')] &= \mathop{\mathrm{E}}_{x,x'}[\Psi(x)^\top \Psi(x')\Psi'(x')^\top \Psi'(x)] \\
&= \mathop{\mathrm{E}}_{x,x'}\big[\operatorname{Tr}[\Psi(x)^\top \Psi(x')\Psi'(x')^\top \Psi'(x)]\big] \\
&= \langle \mathop{\mathrm{E}}_{x}[\Psi(x)\Psi'(x)^\top], \mathop{\mathrm{E}}_{x'}[\Psi(x')\Psi'(x')^\top]\rangle_F = \|\mathbf{U}\|_F^2,
\end{aligned}$$

where $\mathbf{U} = \mathrm{E}_x[\Psi(x)\Psi'(x)^\top]$. □

The following similarly defines the alignment between two kernel matrices $\mathbf{K}$ and $\mathbf{K}'$ based on a finite sample $S = (x_1, \ldots, x_m)$ drawn according to D.

Definition 2.4. *Let $\mathbf{K} \in \mathbb{R}^{m\times m}$ and $\mathbf{K}' \in \mathbb{R}^{m\times m}$ be two kernel matrices such that $\|\mathbf{K}_c\|_F \neq 0$ and $\|\mathbf{K}'_c\|_F \neq 0$. Then, the* alignment *between $\mathbf{K}$ and $\mathbf{K}'$ is defined by*

$$\widehat{\rho}(\mathbf{K}, \mathbf{K}') = \frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{\|\mathbf{K}_c\|_F \|\mathbf{K}'_c\|_F} \ .$$

Here too, by the Cauchy-Schwarz inequality, $\widehat{\rho}(\mathbf{K}, \mathbf{K}') \in [-1, 1]$ and in fact $\widehat{\rho}(\mathbf{K}, \mathbf{K}') \geq 0$ since the Frobenius product of any two positive semi-definite matrices $\mathbf{K}$ and $\mathbf{K}'$ is non-negative. Indeed, for such matrices, there exist matrices $\mathbf{U}$ and $\mathbf{V}$ such that $\mathbf{K} = \mathbf{U}\mathbf{U}^\top$ and $\mathbf{K}' = \mathbf{V}\mathbf{V}^\top$. The statement follows from

$$\langle \mathbf{K}, \mathbf{K}' \rangle_F = \mathrm{Tr}(\mathbf{U}\mathbf{U}^\top \mathbf{V}\mathbf{V}^\top) = \mathrm{Tr}\left((\mathbf{U}^\top \mathbf{V})^\top (\mathbf{U}^\top \mathbf{V})\right) \geq 0. \tag{2.40}$$

Our definitions of alignment between kernel functions or between kernel matrices differ from those originally given by Cristianini et al. (2001); Cristianini et al. (2002):

$$A = \frac{\mathrm{E}[KK']}{\sqrt{\mathrm{E}[K^2]\,\mathrm{E}[K'^2]}} \quad \widehat{A} = \frac{\langle \mathbf{K}, \mathbf{K}' \rangle_F}{\|\mathbf{K}\|_F \|\mathbf{K}'\|_F}, \tag{2.41}$$

which are thus in terms of K and K' instead of K_c and K'_c and similarly for matrices. This may appear to be a technicality, but it is in fact a critical difference. Without that centering, the definition of alignment does not correlate well with performance.

To see this, consider the standard case where K' is the target label kernel, that is $K'(x, x') = yy'$, with y the label of x and y' the label of y', and examine the following simple example in dimension two ($\mathcal{X} = \mathbb{R}^2$), where $K(x, x') = x \cdot x' + 1$ and where the distribution, D, is defined by a fraction $\alpha \in [0, 1]$ of all points being at $(-1, 0)$ and labeled with -1, and the remaining points at $(1, 0)$ with label $+1$.

Clearly, for any value of $\alpha \in [0, 1]$, the problem is separable for example with the simple vertical line going through the origin and one would expect the alignment to be 1. However, the alignment A is never equal to one except for $\alpha = 0$ or $\alpha = 1$ and in fact, for the balanced case where $\alpha = 1/2$, its value is $A = 1/\sqrt{2} \approx .707$. In contrast, with our definition, $\rho(K, K') = 1$ for all

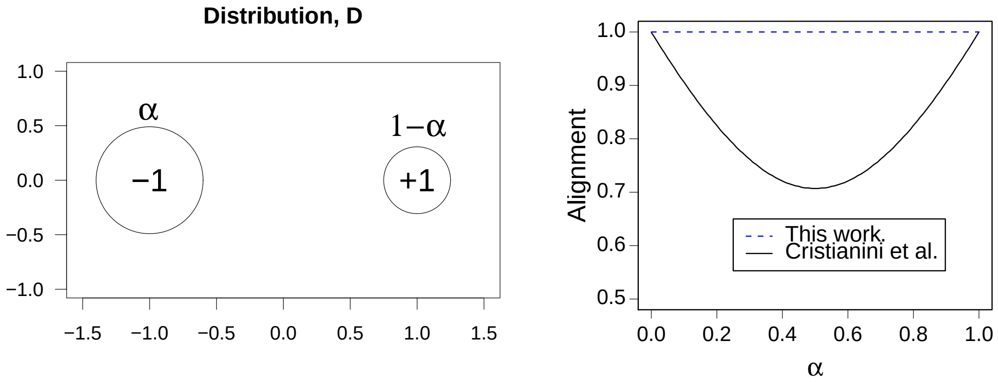


Figure 2.4: Alignment values computed for two different definitions of alignment: $A = [\frac{1+(1-2\alpha)^2}{2}]^{\frac{1}{2}}$ in black, $\rho = 1$ in blue. In this simple two-dimensional example, a fraction α of the points are at $(-1, 0)$ and have the label -1. The remaining points are at $(1, 0)$ and have the label $+1$.

$\alpha \in [0, 1]$, see Figure 2.4.

This mismatch between A (or $\widehat{A}$) and the performance values can also be seen in real world datasets. Instances of this problem have been noticed by Meila (2003) and Pothin and Richard (2008) who have suggested various (input) data translation methods, and by Cristianini et al. (2002) who observed an issue for unbalanced data sets. Table 2.1 also gives a series of empirical results in several tasks illustrating the fact that the quantity $\widehat{A}$ measured with respect to several different kernels does not always correlate well with the performance achieved by each kernel. In fact for the splice dataset, the non-centered alignment is positively correlated with the error-rate, while a large negative correlation is expected of a good quality measure. The centered notion of alignment, $\widehat{\rho}$, however, shows good correlation along all datasets and is always better correlated that $\widehat{A}$. The definitions we are adopting are general

	KINEMATICS	IONOSPHERE	GERMAN	SPAMBASE	SPLICE
$\widehat{\rho}$	-0.9624	-0.9979	-0.9439	-0.9918	-0.9515
$\widehat{A}$	-0.8627	-0.9841	-0.9390	-0.9889	0.4484

Table 2.1: The correlations of the alignment values and error-rates of various kernels. The top row displays the correlation of errors of the base kernels used in Section 3.3.2 with centered alignments ($\widehat{\rho}$) and the bottom row displays the correlation with non-centered alignment ($\widehat{A}$).

and require centering for both kernels K and K'.

The notion of alignment seeks to capture the correlation between the random variables $K(x, x')$ and $K'(x, x')$ and one could think it natural, as for the standard correlation coefficients, to consider the following definition:

$$\rho'(K, K') = \frac{\mathrm{E}[(K - \mathrm{E}[K])(K' - \mathrm{E}[K'])]}{\sqrt{\mathrm{E}[(K - \mathrm{E}[K])^2]\,\mathrm{E}[(K' - \mathrm{E}[K'])^2]}} \ . \qquad (2.42)$$

However, centering the kernel values is not directly relevant to linear predictions in feature space, while our definition of alignment, ρ, is precisely related to that. Also, as already shown in Section 2.5.1, centering in the feature space implies the centering of the kernel values, since $\mathrm{E}[K_c] = 0$ and $\frac{1}{m^2}\sum_{i,j=1}^{m}[\mathbf{K}_c]_{ij} = 0$ for any kernel K and kernel matrix $\mathbf{K}$. Conversely, however, centering of the kernel does not imply centering in feature space.

This section establishes several important properties of the alignments ρ and its empirical estimate $\widehat{\rho}$: we give a concentration bound of the form $|\rho - \widehat{\rho}| \leq O(1/\sqrt{m})$, and show the existence of good prediction hypotheses both for classification and regression, in the presence of high alignment.

2.5.2 Concentration Bound

Our concentration bound differs from that of Cristianini et al. (2001) both because our definition of alignment is different and because we give a bound directly on the quantity of interest $|\rho - \widehat{\rho}|$. Instead, Cristianini et al. give a bound on $|A' - \widehat{A}|$, where

$$A' = \frac{\mathrm{E}_S[\sum_{i,j=1}^m K(x_i, x_j)K'(x_i, x_j)]}{\sqrt{\mathrm{E}_S[\sum_{i,j=1}^m K(x_i, x_j)^2]\,\mathrm{E}_S[\sum_{i,j=1}^m K'(x_i, x_j)^2]}}. \tag{2.43}$$

Thus, $A' \neq A$ can be related to A by replacing each Frobenius product with its expectation over samples of size m. When compared to A, which takes an expectation over independent pairs of points, A' has a strong diagonal bias. That is, at least m of the terms in the expectation will be of the form $K(x_i, x_i)K'(x_i, x_i)$. It can be shown that A' will converge to the correct value of A as m increases, but to best of our knowledge this had not been carefully studied previously. In the appendix, lemma A.5 bounds exactly the difference between an expectation based on independent pairs of points and the expectation based on a sample of size m.

Using this we can give a bound on the essential quantities appearing in the definition of the alignments. The proof and relative supporting lemmas are found in the appendix.

Proposition 2.6. *Let* $\mathbf{K}$ *and* $\mathbf{K}'$ *denote kernel matrices associated to the kernel functions* K *and* K' *for a sample of size* m *drawn according to* D.

Assume that for any $x \in \mathcal{X}$, $K(x, x) \leq R^2$ and $K'(x, x) \leq R^2$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following inequality holds:

$$\left| \frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2} - \mathrm{E}[K_c K'_c] \right| \leq \frac{18R^4}{m} + 24R^4 \sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$

Theorem 2.10. *Under the assumptions of Proposition 2.6, and further assuming that the conditions of the Definitions 2.3-2.4 are satisfied for $\rho(K, K')$ and $\widehat{\rho}(\mathbf{K}, \mathbf{K}')$, for any $\delta > 0$, with probability at least $1 - \delta$, the following inequality holds:*

$$|\rho(K, K') - \widehat{\rho}(\mathbf{K}, \mathbf{K}')| \leq 18\beta \left[\frac{3}{m} + 4\sqrt{\frac{\log \frac{6}{\delta}}{2m}} \right],$$

with $\beta = \max(R^4 / \mathrm{E}[K_c^2], R^4 / \mathrm{E}[K_c'^2])$.

Proof. To shorten the presentation, we first simplify the notation for the alignments as follows:

$$\rho(K, K') = \frac{b}{\sqrt{aa'}} \qquad \widehat{\rho}(\mathbf{K}, \mathbf{K}') = \frac{\widehat{b}}{\sqrt{\widehat{a}\widehat{a}'}},$$

with $b = \mathrm{E}[K_c K'_c]$, $a = \mathrm{E}[K_c^2]$, $a' = \mathrm{E}[K_c'^2]$ and similarly, $\widehat{b} = (1/m^2)\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F$, $\widehat{a} = (1/m^2)\|\mathbf{K}_c\|^2$, and $\widehat{a}' = (1/m^2)\|\mathbf{K}'_c\|^2$. By Proposition 2.6 and the union bound, for any $\delta > 0$, with probability at least $1-\delta$, all three differences $a - \widehat{a}$, $a' - \widehat{a}'$, and $b - \widehat{b}$ are bounded by $\alpha = \frac{18R^4}{m} + 24R^4 \sqrt{\frac{\log \frac{6}{\delta}}{2m}}$. Using the definitions of ρ and $\widehat{\rho}$, we can write:

$$\begin{aligned}|\rho(K,K') - \widehat{\rho}(\mathbf{K},\mathbf{K}')| &= \Big|\frac{b}{\sqrt{aa'}} - \frac{\widehat{b}}{\sqrt{\widehat{a}\widehat{a}'}}\Big| = \Big|\frac{b\sqrt{\widehat{a}\widehat{a}'} - \widehat{b}\sqrt{aa'}}{\sqrt{aa'\widehat{a}\widehat{a}'}}\Big| \\ &= \Big|\frac{(b-\widehat{b})\sqrt{\widehat{a}\widehat{a}'} - \widehat{b}(\sqrt{aa'} - \sqrt{\widehat{a}\widehat{a}'})}{\sqrt{aa'\widehat{a}\widehat{a}'}}\Big| \\ &= \Big|\frac{(b-\widehat{b})}{\sqrt{aa'}} - \widehat{\rho}(\mathbf{K},\mathbf{K}')\frac{aa' - \widehat{a}\widehat{a}'}{\sqrt{aa'}(\sqrt{aa'} + \sqrt{\widehat{a}\widehat{a}'})}\Big|.\end{aligned}$$

Since $\widehat{\rho}(\mathbf{K},\mathbf{K}') \in [0,1]$, it follows that

$$|\rho(K,K') - \widehat{\rho}(\mathbf{K},\mathbf{K}')| \leq \frac{|b-\widehat{b}|}{\sqrt{aa'}} + \frac{|aa' - \widehat{a}\widehat{a}'|}{\sqrt{aa'}(\sqrt{aa'} + \sqrt{\widehat{a}\widehat{a}'})}.$$

Assume first that $\widehat{a} \leq \widehat{a}'$. Rewriting the right-hand side to make the differences $a - \widehat{a}$ and $a' - \widehat{a}'$ appear, we obtain:

$$\begin{aligned}|\rho(K,K') - \widehat{\rho}(\mathbf{K},\mathbf{K}')| &\leq \frac{|b-\widehat{b}|}{\sqrt{aa'}} + \frac{|(a-\widehat{a})a' + \widehat{a}(a' - \widehat{a}')|}{\sqrt{aa'}(\sqrt{aa'} + \sqrt{\widehat{a}\widehat{a}'})} \\ &\leq \frac{\alpha}{\sqrt{aa'}}\left[1 + \frac{a' + \widehat{a}}{\sqrt{aa'} + \sqrt{\widehat{a}\widehat{a}'}}\right] \\ &\leq \frac{\alpha}{\sqrt{aa'}}\left[1 + \frac{a'}{\sqrt{aa'}} + \frac{\widehat{a}}{\sqrt{\widehat{a}\widehat{a}'}}\right] \\ &\leq \frac{\alpha}{\sqrt{aa'}}\left[2 + \sqrt{\frac{a'}{a}}\right] = \left[\frac{2}{\sqrt{aa'}} + \frac{1}{a}\right]\alpha.\end{aligned}$$

We can similarly obtain $\left[\frac{2}{\sqrt{aa'}} + \frac{1}{a'}\right]\alpha$ when $\widehat{a}' \leq \widehat{a}$. Both bounds are less than or equal to $3\max(\frac{\alpha}{a}, \frac{\alpha}{a'})$. □

2.5.3 Existence of Good Predictors

For classification and regression tasks, the target kernel is based on the labels and defined by $K_Y(x, x') = yy'$, where we denote by y the label of point x and y' that of x'. This section shows the existence of predictors with high accuracy both for classification and regression when the alignment $\rho(K, K_Y)$ between the kernel K and K_Y is high.

In the regression setting, we shall assume that the labels have been first normalized by dividing by the standard deviation (assumed finite), $\mathrm{E}[y^2] = 1$. In the classification setting we have $y = \pm 1$ and thus we also have $\mathrm{E}[y^2] = 1$. Let h^* denote the hypothesis defined for all $x \in \mathcal{X}$ by

$$h^*(x) = \frac{\mathrm{E}_{x'}[y' K_c(x, x')]}{\sqrt{\mathrm{E}[K_c^2]}}. \tag{2.44}$$

Observe that by definition of h^*, $\mathrm{E}_x[yh^*(x)] = \rho(K, K_Y)$. For any $x \in \mathcal{X}$, define $\gamma(x) = \sqrt{\frac{\mathrm{E}_{x'}[K_c^2(x,x')]}{\mathrm{E}_{x,x'}[K_c^2(x,x')]}}$ and $\Gamma = \max_x \gamma(x)$. The following result shows that the hypothesis h^* has high accuracy when the kernel alignment is high and Γ not too large.[1]

Theorem 2.11 (classification)**.** *Let $R(h^*) = \Pr[yh^*(x) < 0]$ denote the error of h^* in binary classification. For any kernel K such that $0 < \mathrm{E}[K_c^2] < +\infty$, the following holds:*

$$R(h^*) \leq 1 - \rho(K, K_Y)/\Gamma.$$

[1] A version of this result was presented by Cristianini et al. (2001); Cristianini et al. (2002) for the so-called Parzen window solution and non-centered kernels, but their proof implicitly relies on the fact that $\max_x \left[\frac{\mathrm{E}_{x'}[K^2(x,x')]}{\mathrm{E}_{x,x'}[K^2(x,x')]}\right]^{\frac{1}{2}} = 1$ which holds only if K is constant.

Proof. Note that for all $x \in \mathcal{X}$,

$$\begin{aligned}
|yh^*(x)| &= \frac{|y\,\mathrm{E}_{x'}[y'K_c(x,x')]|}{\sqrt{\mathrm{E}[K_c^2]}} \\
&\leq \frac{\sqrt{\mathrm{E}_{x'}[y'^2]\,\mathrm{E}_{x'}[K_c^2(x,x')]}}{\sqrt{\mathrm{E}[K_c^2]}} \\
&= \frac{\sqrt{\mathrm{E}_{x'}[K_c^2(x,x')]}}{\sqrt{\mathrm{E}[K_c^2]}} \leq \Gamma.
\end{aligned}$$

In view of this inequality, and the fact that $\mathrm{E}_x[yh^*(x)] = \rho(K, K_Y)$, we can write:

$$\begin{aligned}
1 - R(h^*) = \Pr[yh^*(x) \geq 0] &= \mathrm{E}[\mathbf{1}_{\{yh^*(x)\geq 0\}}] \\
&\geq \mathrm{E}[\frac{yh^*(x)}{\Gamma}\mathbf{1}_{\{yh^*(x)\geq 0\}}] \\
&\geq \mathrm{E}[\frac{yh^*(x)}{\Gamma}] = \rho(K, K_Y)/\Gamma,
\end{aligned}$$

where $\mathbf{1}_\omega$ is the indicator variable of an event ω. □

A probabilistic version of the theorem can be straightforwardly derived by noting that by Markov's inequality, for any $\delta > 0$, with probability at least $1 - \delta$, $|\gamma(x)| \leq 1/\sqrt{\delta}$.

Theorem 2.12 (regression)**.** *Let* $R(h^*) = \mathrm{E}_x[(y - h^*(x))^2]$ *denote the error of* h^* *in regression. For any kernel* K *such that* $0 < \mathrm{E}[K_c^2] < +\infty$, *the following holds:*

$$R(h^*) \leq 2(1 - \rho(K, K_Y)).$$

Proof. By the Cauchy-Schwarz inequality, it follows that:

$$\begin{aligned}
\operatorname*{E}_x[h^{*2}(x)] &= \operatorname*{E}_x\left[\frac{\mathrm{E}_{x'}[y'K_c(x,x')]^2}{\mathrm{E}[K_c^2]}\right] \\
&\le \operatorname*{E}_x\left[\frac{\mathrm{E}_{x'}[y'^2]\,\mathrm{E}_{x'}[K_c^2(x,x')]}{\mathrm{E}[K_c^2]}\right] \\
&= \frac{\mathrm{E}_{x'}[y'^2]\,\mathrm{E}_{x,x'}[K_c^2(x,x')]}{\mathrm{E}[K_c^2]} = \operatorname*{E}_{x'}[y'^2] = 1.
\end{aligned}$$

Using again the fact that $\mathrm{E}_x[yh^*(x)] = \rho(K, K_Y)$, the error of h^* can be bounded as follows:

$$\begin{aligned}
\mathrm{E}[(y-h^*(x))^2] &= \operatorname*{E}_x[h^*(x)^2] + \operatorname*{E}_x[y^2] - 2\operatorname*{E}_x[yh^*(x)] \\
&\le 1 + 1 - 2\rho(K, K_Y). \qquad \square
\end{aligned}$$

2.5.4 Unnormalized Alignment

Note, a simpler notion of alignment, as well as accompanying bounds, is possible if we consider the unnormalized alignment, denoted by η.

Definition 2.5. *Let K and K' be two kernel functions defined over $\mathcal{X} \times \mathcal{X}$. Then, the* unnormalized alignment *between K and K' is defined by,*

$$\eta(K, K') = \operatorname*{E}_{x,x'}[K_c(x,x')K'_c(x,x')] \ .$$

Similarly, the empirical unnormalized alignment is denoted $\widehat{\eta}$ and is defined as the Frobenius product between two kernel matrices.

Definition 2.6. *Let* $\mathbf{K} \in \mathbb{R}^{m\times m}$ *and* $\mathbf{K}' \in \mathbb{R}^{m\times m}$ *be two kernel matrices. Then, the* unnormalized alignment *between* $\mathbf{K}$ *and* $\mathbf{K}'$ *is defined by,*

$$\widehat{\eta}(\mathbf{K}, \mathbf{K}') = \langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F = \frac{1}{m^2} \sum_{i,j=1}^{m} K_c(x_i, x_j) K'_c(x_i, x_j) \ .$$

Clearly, since the functions η and $\widehat{\eta}$ are unnormalized, their output is no longer guaranteed to be in the interval $[0, 1]$. However, assuming the kernel function K and labels are bounded, the unnormalized alignment between K and K_Y can be bounded as well.

Lemma 2.3. *Let* K *be a kernel function and assume for all* $x \in \mathcal{X}$, $K_c(x, x) \leq R^2$ *and for all* y *we have* $y \leq M$*. Then the following bounds hold,*

$$0 \leq \eta(K, K_Y) \leq R^2 M \,, \qquad\qquad 0 \leq \widehat{\eta}(\mathbf{K}, \mathbf{K}_Y) \leq R^2 M \,.$$

Proof. The lower bounds can be shown as in the case of the normalized alignment ρ, assuming K is PSD. Both bounds are shown via the application of Cauchy-Schwarz and using the fact $\sup_{y,y'}(yy')^2 \leq M^2$ and $\sup_{x,x'} K^2(x, x') \leq \sup_x K^2(x, x) \leq R^4$. For the first inequality we have,

$$\begin{aligned}
\eta^2(K, K_Y) &= \mathop{\mathrm{E}}_{(x,y),(x',y')}[K_c(x, x')yy']^2 \\
&\leq \mathop{\mathrm{E}}_{x,x'}[K_c^2(x, x')] \mathop{\mathrm{E}}_{y,y'}[yy']^2 \\
&\leq R^4 M^2 \,.
\end{aligned}$$

Similarly, for the second inequality,

$$\begin{aligned}
\widehat{\eta}(\mathbf{K}, \mathbf{K}') &= \frac{1}{m^2} \sum_{i,j=1}^{m} K_c(x_i, x_j) y_i y_j \\
&\leq \frac{1}{m^2} \sqrt{\sum_{i,j=1}^{m} K_c^2(x_i, x_j)} \sqrt{\sum_{i,j=1}^{m} (y_i y_j)^2} \\
&\leq \frac{1}{m^2} \sqrt{m^2 R^4} \sqrt{m^2 M^2} \leq R^2 M \, .
\end{aligned}$$

□

Note in the case of classification we simply have $M = 1$. Straight-forward bounds can be given for a "good" hypothesis, without the dependence on a term such as Γ. Here we define a good hypothesis as follows,

$$g^*(x) = \mathop{\mathrm{E}}_{x'}[y' K_c(x, x')] \, . \tag{2.45}$$

Below we show a classification bound that depends on the unnormalized alignment.

Theorem 2.13 (classification)**.** *Let $R(g^*) = \Pr[y g^*(x) < 0]$ denote the error of g^* in binary classification. For any kernel K such that $\sup_{x \in \mathcal{X}} K_c(x, x) \leq R^2$, we have:*

$$R(g^*) \leq 1 - \eta(K, K_Y)/R^2.$$

Proof. Note that for all $x \in \mathcal{X}$,

$$|yg^*(x)| = |g^*(x)| = |\operatorname*{E}_{x'}[y'K_c(x, x')]| \leq R^2.$$

Using this inequality, and the fact that $\mathrm{E}_x[yg^*(x)] = \eta(K, K_Y)$, we can write:

$$\begin{aligned} 1 - R(g^*) &= \Pr[yg^*(x) \geq 0] = \mathrm{E}[\mathbf{1}_{\{yg^*(x) \geq 0\}}] \\ &\geq \mathrm{E}[\frac{yg^*(x)}{R^2}\mathbf{1}_{\{yh^*(x) \geq 0\}}] \geq \mathrm{E}[\frac{yg^*(x)}{R^2}] \\ &= \eta(K, K_Y)/R^2, \end{aligned}$$

where $\mathbf{1}_\omega$ is the indicator variable of an event ω. □

Although the unnormalized alignment, η, allows for a simpler analysis (in the classification setting) it may also suffer from an unfair bias when comparing two kernels with very different norms. For this reason, in practice, it would be best to first initially normalize each kernel that is being compared. This is exactly what is done in Section 3.3.1 when deriving the independent alignment-based weighting, which maximizes the unnormalized alignment.

The theoretical results which have been shown in this section help motivate the algorithms of Section 3.3. It will be confirmed empirically that simpler two-stage methods can, in fact, outperform global one-stage methods in several tasks.

Chapter 3

Algorithms and Empirical Results

3.1 Previous Results

In this section we focus on algorithms used in learning linear combinations of base kernels. This family of kernels was first popularized in Lanckriet et al. (2002); Lanckriet et al. (2004b), and has become one of the most studied and most principled families of kernels used for the automatic kernel selection problem. In their seminal work, the choice of kernel is based on the SVM objective,

$$\begin{aligned} \min_{K \in \mathcal{K}} \max_{\boldsymbol{\alpha}} \ & \sum_{i=1}^{m} \alpha_i - \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \text{subject to } & \sum_{i=1}^{m} \alpha_i y_i = 0 \ \wedge \ \forall i, 0 \leq \alpha_i \leq C \end{aligned} \tag{3.1}$$

In this sense, both the kernel and hypothesis are selected in a single global optimization problem. Two families of linear combinations were considered, one with a PDS constraint and another with a positivity constraint, which is sufficient to ensure that the resulting kernel is PDS, but also more restrictive,

$$\mathcal{K}_+ = \{K_{\boldsymbol{\mu}} = \sum_{k=1}^{p} \mu_k K_k : K_{\boldsymbol{\mu}} \succeq 0, \|\boldsymbol{\mu}\|_1 \leq \Lambda\}\,, \tag{3.2}$$

$$\mathcal{K}_{++} = \{K_{\boldsymbol{\mu}} = \sum_{k=1}^{p} \mu_k K_k : \boldsymbol{\mu} \succeq 0, \|\boldsymbol{\mu}\|_1 \leq \Lambda\}\,. \tag{3.3}$$

Both families of kernels enforce a restriction on the L_1-norm of the weight vector, $\boldsymbol{\mu}$, which acts as regularization, and that also encourages a sparse weight vector solution. In order to solve the optimization problem using the more general family of kernels, $\mathcal{K}_+$, the authors solve a semi-definite programming (SDP) problem. If the more restrictive class, $\mathcal{K}_{++}$, is used, then the problem can be reduced to a quadratically constrained quadratic program (QCQP), which is much more efficiently solved in practice. It is shown experimentally, that using the class $\mathcal{K}_{++}$ instead of $\mathcal{K}_+$ does not result any notable difference in performance. In addition to solving the problem using for the SVM objective, the authors also present similar algorithms for solving the kernel ridge regression (KRR) algorithm.

Several other methods have been suggested to solve the same problem, using various different optimization formulations. The problem can also be solved in an sequential minimization optimization (SMO) type approach (Bach et al., 2004), semi-infinite linear program (SILP) optimization (Sonnenburg et al., 2006), subgradient optimization (Rakotomamonjy et al., 2008) or bundle method (Xu et al., 2009). The goal of all these methods are to provide a more efficient algorithm, while the solution of this convex problem is of course meant to be same the in all cases.

To study the effectiveness of these algorithms, Lanckriet et al. (2004a) present experiments using several publicly available datasets that primarily use families of Gaussian kernels with varying bandwidth or polynomial kernels of varying degrees as base kernels. It is demonstrated that is such cases, solving the problem in (3.1) provides performance that is very comparable to the performance that is achieved by doing a more costly cross-validation to find a single best kernel. Furthermore, results are also shown for datasets which combine kernels based on heterogeneous sources of data. In this case, the learned combination often performs even better than any single base kernel. Despite these successes, this initial paper contained no comparisons to a very simple yet effective baseline: a *uniform* combination of base kernels. In several settings, as described in this thesis, as well as observed in other settings (Cortes, 2009; Cortes et al., 2008a), this very simple baseline often performs comparably to the much more computationally complex methods used to learn combination weights.

Finally, we note that there has also been some very recent developments in learning non-linear combinations (Varma & Babu, 2009; Bach, 2008; Cortes et al., 2009b), although the problem has been less studied. Furthermore, in the most general case, the associated optimization problem for maximizing the SVM or KRR objective becomes non-convex and the global optimum solution cannot be guaranteed.

It will be the focus of this chapter to show different algorithms and approaches for improving the performance of linear combinations of kernels, compared to both the algorithm suggested in (3.1) as well as the surprisingly effective baseline of uniform combinations.

Section 3.2 will present algorithms and results for L_2 regularized families of kernels that encourage non-sparse optimal combinations of base kernels. Not only does the proposed method improve over the baseline uniform combination in certain scenarios, but also demonstrates a sparse L_1 type regularization may not always be helpful.

Then in Section 3.3, we explore two-stage kernel selection algorithms, where the selection of the kernel and hypothesis are separate. Here, we show that solving two simpler problems with fewer parameters can lead to improvements over the single-stage algorithms that have been investigated previously and also lead to improvements over the uniform baseline in several tasks.

3.2 Non-Sparse Regularization

As mentioned in the previous section, a common family of kernels considered is that of non-negative combinations of some fixed kernels constrained by an L_1 regularization. This section studies the problem of learning kernels using a linear family of kernels but with an L_2 regularization instead. Note that an L_2 regularization, unlike the L_1 regularization, provides a non-sparse weight vector. This type of solution may be desirable in cases where the number of base kernels is large and each one is generated from a different set of features. In such a setting it may be undesirable to "throw away" any kernels (or equivalently features) by setting its corresponding weight to zero. Similar observations are made in work by Kloft et al. (2009). Experiments carried out with a number of datasets, including those used by previous authors (Lanckriet et al., 2004a; Cortes et al., 2008b), in some of which using an L_2 regularization turned out to be significantly beneficial and otherwise never worse than using L_1 regularization. We report these results in the experimental section.

3.2.1 Solving the LKRR Problem

In this section we examine the performance of L_2-regularized kernel-learning on a number of datasets.

Problem (2.19) is a convex optimization problem and can thus be solved using standard gradient descent-type algorithms. However, the form of the solution provided by Theorem 2.1, $\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1}$, motivates an iterative algorithm that proved to be significantly faster in our experiments. The following gives the pseudo-code of the algorithm, where $\eta \in (0, 1)$ is an interpolation parameter and $\epsilon > 0$ a convergence error. In our experiments, the number of iterations needed on average for convergence was about 10 to 15. When using a small number of kernels with few data points, each iteration took a fraction of a second, while when using thousands of kernels and data-points each iteration took about a second.

Algorithm 1 Interpolated Iterative Algorithm

```
Input: $\mathbf{K}_k$, $k \in [1, p]$
$\boldsymbol{\alpha}' \leftarrow (\mathbf{K}_0 + \lambda\mathbf{I})^{-1}\mathbf{y}$
repeat
  $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}'$
  $\mathbf{v} \leftarrow (\boldsymbol{\alpha}^\top K_1 \boldsymbol{\alpha}, \ldots, \boldsymbol{\alpha}^\top K_p \boldsymbol{\alpha})^\top$
  $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}_0 + \Lambda \frac{\mathbf{v}}{\|\mathbf{v}\|}$
  $\boldsymbol{\alpha}' \leftarrow \eta\boldsymbol{\alpha} + (1 - \eta)(\mathbf{K}(\boldsymbol{\alpha}) + \lambda\mathbf{I})^{-1}\mathbf{y}$
until $\|\boldsymbol{\alpha}' - \boldsymbol{\alpha}\| < \epsilon$
```

3.2.2 Experimental Results

We did two series of experiments. First, we validated our experimental set-up and our implementation for Algorithm 1 and previous algorithms for L_1 regularization by comparing our results against those previously presented by Lanckriet et al. (2004a), which use a small number of base kernels and relatively small data sets. We then focused on a larger task consisting of learning sequence kernels using thousands of base kernels as described by Cortes et al. (2008b).

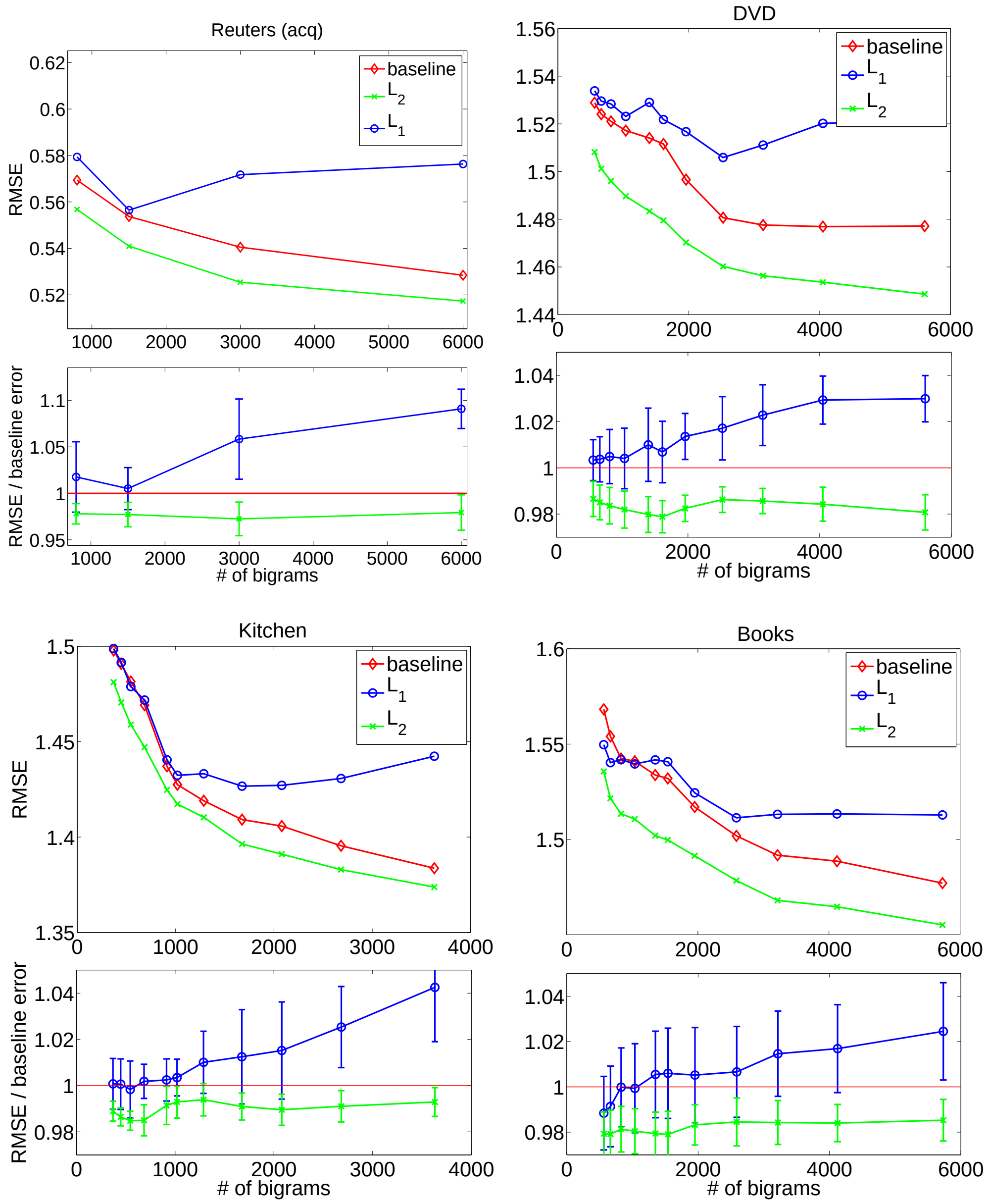


Figure 3.1: RMSE error reported for the Reuters and various sentiment analysis datasets (kitchen, dvds and electronics). The upper plots show the absolute error, while the bottom plots show the error after normalizing by the baseline error (error bars are ±1 standard deviation).

UCI datasets

To verify our implementation, we first evaluated Algorithm 1 on the *breast*, *ionosphere*, *sonar* and *heart* datasets from the UCI ML Repository which were previously used for experimentation by Lanckriet et al. (2004a). In order to use KRR for the classification datasets, we train with ±1 labels and examined both root mean squared error (RMSE) with respect to these target values and the misclassification rate when using the sign of the learned function to classify the test set. We found that both measures of error give similar comparative results. We use exactly the same experimental setup as (Lanckriet et al., 2004a), with three kernels: a Gaussian, a linear, and a second degree polynomial kernel.

For comparison, we consider the best performing single kernel of these three kernels, the performance of an evenly-weighted sum of the kernels, and the performance of an L_1-regularized algorithm (similar to that of Lanckriet et al. (2004a), however using the KRR objective).

Our results on these datasets validate our implementations by reaffirming the results from Lanckriet et al. (2004a). Using kernel-learning algorithms (whether L_1 or L_2 regularized) never does worse than selecting the best single kernel via costly cross-validation. However, our experiments also confirm the findings by Lanckriet et al. (2004a) that kernel-learning algorithms for this setting never do significantly *better*. All differences are easily within one standard deviation, with absolute misclassification rate of: 0.03 (*breast*), 0.08 (*ionosphere*), 0.16 (*sonar*) and 0.17 (*heart*). As our next set of experiments will show, when the number of base kernels is substantially increased, this picture changes completely. The performance of the L_2 regularized kernel is significantly better than the baseline of evenly-weighted sum of kernels, that in turn performs significantly better than the L_1 regularized kernel.

Sequence-based datasets

In our next experiments, we also make use of one of the datasets from (Lanckriet et al., 2004a), the ACQ task of the Reuters-21578 dataset, though we learn with different base kernels. Using the ModApte split we produce 3,299 test examples and 9,603 training examples from which we randomly subsample 2,000 points to train with over 20 trials.

For features we use the N most frequently occurring bigrams, where N is indicated in Figure 3.1. As suggested in Cortes et al. (2008b), we use N rank-1 base kernels, with each kernel corresponding to a particular n-gram. Thus, if $\mathbf{v}_i \in \mathbb{R}^m$ is the vector of the occurrences of the ith n-gram across the training data, then the ith base kernel matrix is defined as $K_i = \mathbf{v}_i \mathbf{v}_i^\top$. Note that these base kernels are orthogonal, since each Φ_i is the projection onto a single distinct component of Φ. The parameters λ and Λ are chosen via 10-fold cross validation on the training data and the $\boldsymbol{\mu}_0$ is set the uniform value $\frac{\Lambda}{\sqrt{p}}$.

We compare the presented L_2-regularized algorithm to both a baseline of the evenly-weighted sum of all the base kernels, as well as to the L_1-regularized method of Cortes et al. (2008b) (Figure 3.1). The results illustrate that for large-scale kernel-learning, kernel selection with L_2 regularization improves performance, and that L_1 regularization can in fact be harmful. Note, that all base kernels here represent orthogonal features, thus, a sparse solution that eliminates a subset of the base kernels may negatively impact performance. Since Lanckriet et al. (2004a) do not perform learning for a large number of base kernels, we cannot directly compare results for this task. However, the best error rate we obtain by classifying the test set by the sign of the L_1-regularized learner is comparable to that reported by Lanckriet et al. (2004a).

For our last experiments we consider the task of sentiment analysis of reviews within several domains: books, dvds, and kitchen appliances (Blitzer et al., 2007). Each domain consists of 2,000 product reviews, each with a rating between 1 and 5. We create 10 random 50/50 splits of the data into a training and test set. For features we again use the N most frequently occurring bigrams and for basis kernels again use N rank-1 kernels, see Figure 3.1. The results on these dataset amplify the result from the Reuters ACQ dataset: L_1 regularization can negatively impact the performance for large number of kernels, while L_2-regularization improve the performance significantly over the baseline over the evenly-weighted sum of kernels.

3.3 Two-Stage Algorithms

This section explores *two-stage* algorithms for learning kernels, where the first stage consists of *learning* a kernel K that is a convex combination of p kernels and the second stage consists of using K with a standard kernel-based learning algorithm such as support vector machines (SVMs) (Cortes & Vapnik, 1995) for classification, or KRR for regression, to select a prediction hypothesis. We show that with this two-stage method it is possible to obtain better performance than with the one-stage methods on several datasets, as well as the baseline uniform combination of kernels.

The criteria used for selecting a kernel is based on the natural notion of *kernel alignment* introduced by Cristianini et al. (2001), though our definition differs from the original one. We note that other measures of similarity could be used in this context. In particular, the notion of similarity suggested by Balcan and Blum (2006) could be used if it could be computed from finite samples.

We present a number of novel algorithmic, and empirical results for the alignment-based two-stage techniques. We give an algorithm for learning a maximum alignment kernel and prove that the mixture coefficients can be obtained efficiently by solving a simple quadratic program (QP) in the case of a convex combination, and even give a closed-form solution in the case of an arbitrary linear combination. We finally report the results of extensive experiments with this alignment-based method both in classification and regression, and compare our results with L_1 and L_2 regularized learning kernel algorithms (Lanckriet et al., 2004a; Cortes et al., 2009a), as well as with the uniform kernel combination method. The results show an improvement both over the uniform combination and over the one-stage kernel learning algorithms in all datasets. We also observe a correlation between the alignment achieved and performance.

3.3.1 Algorithms

This section discusses two-stage algorithms for learning kernels in the form of linear combinations of p base kernels K_k, $k \in [1, p]$. In all cases, the final hypothesis learned belongs to the reproducing kernel Hilbert space associated to a kernel $K_{\boldsymbol{\mu}} = \sum_{k=1}^{p} \mu_k K_k$, where the mixture weights are selected subject to the condition $\boldsymbol{\mu} \succeq \mathbf{0}$, which guarantees that K is a PSD kernel, and a condition on the norm of $\boldsymbol{\mu}$, $\|\boldsymbol{\mu}\| = \Lambda > 0$, where Λ is a regularization parameter.

In the first stage, these algorithms determine the mixture weights $\boldsymbol{\mu}$. In the second stage, they train a kernel-based algorithm, e.g., SVMs for classification, or KRR for regression, in combination with the kernel $\mathbf{K}_{\boldsymbol{\mu}}$, to learn a hypothesis h. Thus, the algorithms differ only by the first stage, where $K_{\boldsymbol{\mu}}$ is determined, which we briefly describe.

Uniform combination (`unif`): this is the most straightforward method, which consists of choosing equal mixture weights, thus the kernel matrix used is,

$$\mathbf{K}_{\boldsymbol{\mu}} = \frac{\Lambda}{p} \sum_{k=1}^{p} \mathbf{K}_k \, . \tag{3.4}$$

Nevertheless, improving upon the performance of this method has been surprisingly difficult for standard (one-stage) learning kernel algorithms (Cortes, 2009).

Independent alignment-based method (`align`): this is a simple but efficient method which consists of using the training sample to independently compute the alignment between each kernel matrix $\mathbf{K}_k$ and the target kernel matrix $\mathbf{K}_Y = \mathbf{y}\mathbf{y}^\top$, based on the labels $\mathbf{y}$, and to choose each mixture weight μ_k proportional to that alignment. Thus, the resulting kernel matrix is:

$$\mathbf{K}_{\boldsymbol{\mu}} \propto \sum_{k=1}^{p} \widehat{\rho}(\mathbf{K}_k, \mathbf{K}_Y)\mathbf{K}_k \,. \tag{3.5}$$

Alignment maximization algorithms (`alignf`): the independent alignment-based method ignores the correlation between the base kernel matrices. The alignment maximization method takes these correlations into account. It determines the mixture weights μ_k jointly by seeking to maximize the alignment between the convex combination kernel $\mathbf{K}_{\boldsymbol{\mu}} = \sum_{k=1}^{p} \mu_k \mathbf{K}_k$ and the target kernel $\mathbf{K}_Y = \mathbf{y}\mathbf{y}^\top$, as suggested by Cristianini et al. (2001); Kandola et al. (2002a) and later studied by Lanckriet et al. (2004a) who showed that the problem can be solved as a QCQP.

In what follows, we present even more efficient algorithms for computing the weights μ_k by showing that the problem can be reduced to a simple QP. We also examine the case of a non-convex linear combination, where components of $\boldsymbol{\mu}$ can be negative, and show that the problem then admits a closed-form solution. We start with this linear combination case and partially use that solution to obtain the solution of the convex combination.

Relationship to Ensemble Methods

An alternative two-stage technique consists of first learning a prediction hypothesis h_k using each kernel K_k, and then learning the best linear combination of these hypotheses,

$$h(x) = \sum_{i=1}^{p} \mu_i h_i(x)\,. \tag{3.6}$$

But, such ensemble-based techniques make use of a richer hypothesis space than the one used by learning kernel algorithms such as (Lanckriet et al., 2004a). To see this, note that the final hypothesis is of the form,

$$h(x) = \sum_{i=1}^{p} \mu_i h_i(x) = \sum_{i=1}^{p} \mu_i \sum_{j=1}^{m} \alpha_j^i K_i(x_j, x) = \sum_{j=1}^{m} \sum_{i=1}^{p} \alpha_j^i \mu_i K_i(x_j, x)\,,$$

for some choice of $\boldsymbol{\alpha}^i \in \mathbb{R}^m$ for all $i \in \{1, \ldots, p\}$. This is not necessarily equal to a hypothesis of the form

$$\sum_{j=1}^{m} \alpha_j \sum_{i=1}^{p} \mu_i K_i(x_j, x) = \sum_{j=1}^{m} \alpha_j K_{\boldsymbol{\mu}}(x_j, x)\,, \tag{3.7}$$

for any choice of $\boldsymbol{\alpha} \in \mathbb{R}^m$. Furthermore, the combination weights μ_i are not required to be positive in this case. Theoretical and empirical comparisons between these different classes of hypotheses may be interesting future work.

Independent alignment-based derivation

One way to understand the suggested independent alignment-based algorithm, is to see it as optimizing the unnormalized, η, alignment with respect to an L2-

constraint on the combination weights and with base kernels that have been normalized with respect to the Frobenius norm.

For completeness, we will analyze the following optimization problem for any $q > 1$,

$$\begin{aligned} \max_{\mu} \quad & \widehat{\eta}(\sum_{k=1}^{p} \mu_k \mathbf{K}_k, \mathbf{K}_Y) = \langle \sum_{k=1}^{p} \mu_k \mathbf{K}_k, \mathbf{K}_Y \rangle_F \\ \text{subject to:} \quad & \sum_{i=1}^{p} \mu_k^q \leq \Lambda \,. \end{aligned} \tag{3.8}$$

Note that there is no explicit constraint forcing $\boldsymbol{\mu} \succeq 0$, but the optimal solution found below will in fact satisfy this as long as $\forall k \in \{1, \ldots, p\}, \mathbf{K}_k \succeq 0$.

Proposition 3.1. *The optimal solution $\boldsymbol{\mu}^*$ to the optimization presented in Equation (3.8) takes the following form for any $q > 1$,*

$$\mu_k \propto \langle \mathbf{K}_k, \mathbf{K}_Y \rangle_F^{\frac{1}{q-1}} \,.$$

Proof. The Lagrangian corresponding the optimization (3.8) is defined as follows,

$$L(\boldsymbol{\mu}, \beta) = -\sum_{k=1}^{p} \mu_k \langle \mathbf{K}_k, \mathbf{K}_Y \rangle_F + \beta(\sum_{i=1}^{p} \mu_k^q - \Lambda) \,,$$

where the dual variable β is non-negative. Taking the derivative with respect to

μ_k and setting the derivative to zero, reveals the form of the optimal solution,

$$\frac{\partial L}{\partial \mu_k} = -\langle \mathbf{K}_k, \mathbf{K}_Y \rangle_F + q\beta\mu_k^{q-1} = 0$$
$$\implies \mu_k \propto \langle \mathbf{K}_k, \mathbf{K}_Y \rangle_F^{\frac{1}{q-1}} .$$

□

Thus, for $q = 2$, we have $\mu_k \propto \langle \mathbf{K}_k, \mathbf{K}_Y \rangle_F$ which is exactly the solution suggested in Equation (3.5) modulo normalization by the Frobenius norm of the base matrix.

Note that for $q = 1$, the optimization becomes trivial and can be solved by simply placing all weight on the μ_k with largest coefficient. That is the μ_k that has corresponding $\mathbf{K}_k$ with the largest alignment.

Alignment maximization algorithm - linear combination

We can assume without loss of generality that the centered base kernel matrices $\mathbf{K}_{kc}$ are independent since otherwise we can select an independent subset. This condition ensures that $\|\mathbf{K}_{\boldsymbol{\mu}_c}\|_F > 0$ for arbitrary $\boldsymbol{\mu}$ and that $\widehat{\rho}(\mathbf{K}_{\boldsymbol{\mu}}, \mathbf{y}\mathbf{y}^\top)$ is well defined (Definition 2.4). By Lemma A.3, $\langle \mathbf{K}_{\boldsymbol{\mu}_c}, \mathbf{K}_{Yc} \rangle_F = \langle \mathbf{K}_{\boldsymbol{\mu}_c}, \mathbf{K}_Y \rangle_F$. Thus, since $\|\mathbf{K}_{Yc}\|_F$ does not depend on $\boldsymbol{\mu}$, the alignment maximization can be written as the following optimization problem:

$$\max_{\boldsymbol{\mu}\in\mathcal{M}} \widehat{\rho}(\mathbf{K}_{\boldsymbol{\mu}}, \mathbf{y}\mathbf{y}^\top) = \max_{\boldsymbol{\mu}\in\mathcal{M}} \frac{\langle \mathbf{K}_{\boldsymbol{\mu}_c}, \mathbf{y}\mathbf{y}^\top \rangle_F}{\|\mathbf{K}_{\boldsymbol{\mu}_c}\|_F}, \tag{3.9}$$

where $\mathcal{M} = \{\boldsymbol{\mu} \colon \|\boldsymbol{\mu}\|_2 = 1\}$. Note, we do not need to explicitly constrain $\boldsymbol{\mu} \succeq \mathbf{0}$, since this is guaranteed at the optimum, as is shown in Proposition 3.2. A similar set can be defined via norm-1 instead of norm-2. As we shall see, however, the problem can be solved in the same way in both cases. Note that, by Lemma A.3, $\mathbf{K}_{\boldsymbol{\mu}_c} = \mathbf{U}_m \mathbf{K}_{\boldsymbol{\mu}} \mathbf{U}_m$ with $\mathbf{U}_m = \mathbf{I} - \mathbf{1}\mathbf{1}^\top/m$, thus, $\mathbf{K}_{\boldsymbol{\mu}_c} = \sum_{k=1}^p \mu_k \mathbf{U}_m \mathbf{K}_k \mathbf{U}_m = \sum_{k=1}^p \mu_k \mathbf{K}_{kc}$. Let $\mathbf{a}$ denote the vector $(\langle \mathbf{K}_{1c}, \mathbf{y}\mathbf{y}^\top \rangle_F, \ldots, \langle \mathbf{K}_{p_c}, \mathbf{y}\mathbf{y}^\top \rangle_F)^\top$ and $\mathbf{M}$ the matrix defined by $\mathbf{M}_{kl} = \langle \mathbf{K}_{kc}, \mathbf{K}_{lc} \rangle_F$, for $k, l \in [1, p]$. Note that since the base kernels are assumed independent, the matrix $\mathbf{M}$ is invertible. Also, in view of non-negativity of the Frobenius product of symmetric PSD matrices shown in Section 2.5.1, the entries of $\mathbf{a}$ and $\mathbf{M}$ are all non-negative. Observe also that $\mathbf{M}$ is a symmetric PSD matrix since for any vector $\mathbf{X} = (x_1, \ldots, x_m)^\top \in \mathbb{R}^m$,

$$\begin{aligned}
\mathbf{X}^\top \mathbf{M} \mathbf{X} &= \sum_{k,l=1}^m x_k x_l \operatorname{Tr}[\mathbf{K}_{kc} \mathbf{K}_{lc}] \\
&= \operatorname{Tr}\Big[\sum_{k,l=1}^m x_k x_l \mathbf{K}_{kc} \mathbf{K}_{lc}\Big] \\
&= \operatorname{Tr}\Big[(\sum_{k=1}^m x_k \mathbf{K}_{kc})(\sum_{l=1}^m x_l \mathbf{K}_{lc})\Big] \\
&= \|\sum_{k=1}^m x_k \mathbf{K}_{kc}\|_F^2 \geq 0.
\end{aligned}$$

Proposition 3.2. *The solution $\boldsymbol{\mu}^\star$ of the optimization problem (3.9) is given by $\boldsymbol{\mu}^\star = \frac{\mathbf{M}^{-1}\mathbf{a}}{\|\mathbf{M}^{-1}\mathbf{a}\|}$.*

Proof. The optimal solution $\boldsymbol{\mu}^\star$ is the solution of the following more explicit problem:

$$
\begin{aligned}
\boldsymbol{\mu}^\star &= \underset{\|\boldsymbol{\mu}\|_2=1}{\operatorname{argmax}} \frac{\sum_{k=1}^p \mu_k \langle \mathbf{K}_{kc}, \mathbf{y}\mathbf{y}^\top \rangle_F}{\| \sum_{k=1}^p \mu_k \mathbf{K}_{kc} \|_F} \\
&= \underset{\|\boldsymbol{\mu}\|_2=1}{\operatorname{argmax}} \frac{(\sum_{k=1}^p \mu_k \langle \mathbf{K}_{kc}, \mathbf{y}\mathbf{y}^\top \rangle_F)^2}{\| \sum_{k=1}^p \mu_k \mathbf{K}_{kc} \|_F^2} \\
&= \underset{\|\boldsymbol{\mu}\|_2=1}{\operatorname{argmax}} \ \frac{(\boldsymbol{\mu}^\top \mathbf{a})^2}{\boldsymbol{\mu}^\top \mathbf{M} \boldsymbol{\mu}} \\
&= \underset{\|\boldsymbol{\mu}\|_2=1}{\operatorname{argmax}} \ \frac{\boldsymbol{\mu}^\top \mathbf{a}\mathbf{a}^\top \boldsymbol{\mu}}{\boldsymbol{\mu}^\top \mathbf{M} \boldsymbol{\mu}}.
\end{aligned}
$$

We recognize the general Rayleigh quotient. Let $\boldsymbol{\nu} = \mathbf{M}^{1/2}\boldsymbol{\mu}$ and $\boldsymbol{\nu}^\star = \mathbf{M}^{1/2}\boldsymbol{\mu}^\star$, then, the problem can be rewritten as

$$
\boldsymbol{\nu}^\star = \underset{\|\mathbf{M}^{-1/2}\boldsymbol{\nu}\|_2=1}{\operatorname{argmax}} \frac{\boldsymbol{\nu}^\top \big[\mathbf{M}^{-1/2}\mathbf{a}\mathbf{a}^\top \mathbf{M}^{-1/2}\big] \boldsymbol{\nu}}{\boldsymbol{\nu}^\top \boldsymbol{\nu}}.
$$

Therefore, the solution is

$$
\begin{aligned}
\boldsymbol{\nu}^\star &= \underset{\|\mathbf{M}^{-1/2}\boldsymbol{\nu}\|_2=1}{\operatorname{argmax}} \ \frac{\big[\boldsymbol{\nu}^\top (\mathbf{M}^{-1/2}\mathbf{a})\big]^2}{\|\boldsymbol{\nu}\|_2^2} \\
&= \underset{\|\mathbf{M}^{-1/2}\boldsymbol{\nu}\|_2=1}{\operatorname{argmax}} \left[\left(\frac{\boldsymbol{\nu}}{\|\boldsymbol{\nu}\|} \right)^\top (\mathbf{M}^{-1/2}\mathbf{a}) \right]^2.
\end{aligned}
$$

Thus, $\boldsymbol{\nu}^\star \in \operatorname{Vec}(\mathbf{M}^{-1/2}\mathbf{a})$ with $\|\mathbf{M}^{-1/2}\boldsymbol{\nu}^\star\|_2 = 1$. This yields immediately $\boldsymbol{\mu}^\star = \frac{\mathbf{M}^{-1}\mathbf{a}}{\|\mathbf{M}^{-1}\mathbf{a}\|}$. □

Alignment maximization algorithm - convex combination

In view of the proof of Proposition 3.2, the alignment maximization problem with the set $\mathcal{M}' = \{\|\boldsymbol{\mu}\|_2 = 1 \wedge \boldsymbol{\mu} \geq \mathbf{0}\}$ can be written as

$$\boldsymbol{\mu}^* = \operatorname*{argmax}_{\boldsymbol{\mu} \in \mathcal{M}'} \frac{\boldsymbol{\mu}^\top \mathbf{a}\mathbf{a}^\top \boldsymbol{\mu}}{\boldsymbol{\mu}^\top \mathbf{M} \boldsymbol{\mu}}. \tag{3.10}$$

The following proposition shows that the problem can be reduced to solving a simple QP.

Proposition 3.3. *Let* $\mathbf{v}^\star$ *be the solution of the following QP:*

$$\min_{\mathbf{v} \geq \mathbf{0}} \mathbf{v}^\top \mathbf{M} \mathbf{v} - 2\mathbf{v}^\top \mathbf{a}. \tag{3.11}$$

Then, the solution $\boldsymbol{\mu}^*$ *of the alignment maximization problem (3.10) is given by* $\boldsymbol{\mu}^\star = \mathbf{v}^\star / \|\mathbf{v}^\star\|$.

Proof. Note that the objective function of problem (3.10) is invariant to scaling. The constraint $\|\boldsymbol{\mu}\| = 1$ only serves to enforce $0 < \|\boldsymbol{\mu}\| < +\infty$. Thus, using the same change of variable as in the proof of Proposition 3.2, we can instead solve the following problem from which we can retrieve the solution via normalization:

$$\boldsymbol{\nu}^\star = \operatorname*{argmax}_{\substack{0 < \|\mathbf{M}^{-1/2}\boldsymbol{\nu}\|_2 < +\infty \\ \mathbf{M}^{-1/2}\boldsymbol{\nu} \geq \mathbf{0}}} \left[\frac{\boldsymbol{\nu}}{\|\boldsymbol{\nu}\|} \cdot (\mathbf{M}^{-1/2}\mathbf{a}) \right]^2.$$

Equivalently, we can solve the following problem for any finite $\lambda > 0$:

$$\max_{\substack{\mathbf{M}^{-1/2}\mathbf{u}\geq\mathbf{0}\\ \|\mathbf{u}\|=\lambda}} \left[\mathbf{u}\cdot\mathbf{M}^{-1/2}\mathbf{a}\right]^2.$$

Observe that for $\mathbf{M}^{-1/2}\mathbf{u} \geq \mathbf{0}$ the inner product is non-negative: $\mathbf{u}\cdot\mathbf{M}^{-1/2}\mathbf{a} = \mathbf{M}^{-1/2}\mathbf{u}\cdot\mathbf{a} \geq 0$, since the entries of $\mathbf{a}$ are non-negative. The dot product can be decomposed as follows:

$$\begin{aligned}\mathbf{u}\cdot\mathbf{M}^{-1/2}\mathbf{a} &= -\frac{1}{2}\|\mathbf{u}-\mathbf{M}^{-1/2}\mathbf{a}\|^2 + \frac{1}{2}\|\mathbf{u}\|^2 + \frac{1}{2}\|\mathbf{M}^{-1/2}\mathbf{a}\|^2\\ &= -\frac{1}{2}\|\mathbf{u}-\mathbf{M}^{-1/2}\mathbf{a}\|^2 + \frac{\lambda^2}{2} + \frac{1}{2}\|\mathbf{M}^{-1/2}\mathbf{a}\|^2.\end{aligned}$$

Thus, the problem becomes equivalent to the minimization:

$$\min_{\substack{\mathbf{M}^{-1/2}\mathbf{u}\geq\mathbf{0}\\ \|\mathbf{u}\|=\lambda}} \left\|\mathbf{u}-\mathbf{M}^{-1/2}\mathbf{a}\right\|^2. \tag{3.12}$$

Now, we can omit the condition on the norm of $\mathbf{u}$ since (3.12) holds for arbitrary finite $\lambda > 0$ and since neither $\mathbf{u} = \mathbf{0}$ or any infinite norm $\mathbf{u}$ can be the solution even without this condition. Thus, we can now consider instead:

$$\min_{\mathbf{M}^{-1/2}\mathbf{u}\geq\mathbf{0}} \left\|\mathbf{u}-\mathbf{M}^{-1/2}\mathbf{a}\right\|^2.$$

The change of variable $\mathbf{u} = \mathbf{M}^{1/2}\mathbf{v}$ leads to: $\min_{\mathbf{v}\geq\mathbf{0}} \left\|\mathbf{M}^{1/2}\mathbf{v}-\mathbf{M}^{-1/2}\mathbf{a}\right\|^2$. This is a standard least-square regression problem with non-negativity constraints, a simple and widely studied QP for which several families of algorithms have been designed. Expanding the terms, we obtain the equivalent problem:

$$\min_{\mathbf{v} \geq \mathbf{0}} \mathbf{v}^\top \mathbf{M} \mathbf{v} - 2\mathbf{v}^\top \mathbf{a}\,. \qquad \square$$

Note that this QP problem does not require a matrix inversion of $\mathbf{M}$. Also, it is not hard to see that this problem is equivalent to solving a hard margin SVM problem, thus, any SVM solver can also be used to solve it. A similar problem with the non-centered definition of alignment is treated by Kandola et al. (2002b), but their optimization solution differs from ours by adding an additional regularization term on $\boldsymbol{\mu}$ and requires cross-validation.

3.3.2 Experimental Results

This section compares the performance of several learning kernel algorithms for classification and regression. We compare the algorithms `unif`, `align`, and `alignf`, from Section 3.3.1, as well as the one-stage algorithms of Lanckriet et al. (2004a) (denoted `l1-svm`), which solves the SVM problem with an L_1 regularized combination of kernels, and the LKRR algorithm (denoted `l2-krr`) as described in Section 3.2, which sovles the KRR problem with an L_2 regularized combination of kernels.

In all experiments, the error measures reported are for 5-fold cross validation, where, in each trial, three folds are used for training, one used for validation, and one for testing. For the two-stage methods, the same training

	KINEMATICS	IONOSPHERE	GERMAN	SPAMBASE	SPLICE
m	351	1000	1000	1000	1000
γ	-3, 3	-3, 3	-4, 3	-12, -7	-9, -3
A	.138(.005)	.467(.085)	25.9(1.8)	18.7(2.8)	15.2(2.2)
	.158(.013)	.242(.021)	.089(.008)	.138(.031)	.122(.011)
B	.137(.005)	.457(.085)	26.0(2.6)	20.9(2.80)	15.3(2.5)
	.155(.012)	.248(.022)	.082(.003)	.099(.024)	.105(.006)
C	.125(.004)	.445(.086)	25.5(1.5)	18.6(2.6)	15.1(2.4)
	.173(.016)	.257(.024)	.089(.008)	.140(.031)	.123(.011)
D	.115(.004)	.442(.087)	24.2(1.5)	18.0(2.4)	13.9(1.3)
	.176(.017)	.273(.030)	.093(.009)	.146(.028)	.124(.011)

Regression Classification

Table 3.1: Error measures (top) and alignment values (bottom) for (A) `unif`, (B) one-stage `l2-krr` or `l1-svm`, (C) `align` and (D) `alignf` with kernels built from linear combinations of Gaussian base kernels. The choice of γ_0, γ_1 is listed in row labeled γ, and m is the size of the dataset used. Shown with ± 1 standard deviation (in parentheses) measured by 5-fold cross-validation.

and validation data is used for both stages of the learning. The regularization parameter Λ is chosen via a grid search based on the performance on the validation set, while the regularization parameter λ is fixed since only the ratio λ/Λ matters. The $\boldsymbol{\mu}_0$ parameter is set to zero for the general kernel combinations, and is chosen to be uniform for the rank-1 kernel combinations.

General kernel combinations

In the first set of experiments, we consider combinations of Gaussian kernels of the form $\mathbf{K}_\gamma(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$, with varying bandwidth parameter $\gamma \in \{2^{\gamma_0}, 2^{\gamma_0+1}, \ldots, 2^{1-\gamma_1}, 2^{\gamma_1}\}$. The values γ_0 and γ_1 are chosen such that the base kernels are sufficiently different in alignment and performance. Each base

	BOOKS	DVD	ELEC	KITCHEN
unif	1.442 ± .015	1.438 ± .033	1.342 ± .030	1.356 ± .016
	.029 ± .005	.029 ± .005	.038 ± .002	.039 ± .006
l2-krr	1.414 ± .020	1.420 ± .034	1.318 ± .031	1.332 ± .016
	.031 ± .004	.031 ± .005	.042 ± .003	.044 ± .007
align	1.401 ± .035	1.414 ± .017	1.308 ± .033	1.312 ± .012
	.046 ± .006	.047 ± .005	.065 ± .004	.076 ± .008

Regression

	BOOKS	DVD	ELEC	KITCHEN
unif	25.8 ± 1.7	24.3 ± 1.5	18.8 ± 1.4	20.1 ± 2.0
	.030 ± .004	.030 ± .005	.040 ± .002	.039 ± .007
l1-svm	28.6 ± 1.6	29.0 ± 2.2	23.8 ± 1.9	23.8 ± 2.2
	.029 ± .012	.038 ± .011	.051 ± .004	.060 ± .006
align	24.3 ± 2.0	21.4 ± 2.0	16.6 ± 1.6	17.2 ± 2.2
	.043 ± .003	.045 ± .005	.063 ± .004	.070 ± .010

Classification

Table 3.2: The error measures (top) and alignment values (bottom) for kernels built with rank-1 feature based kernels on four domain sentiment analysis domains. Shown with ±1 standard deviation as measured by 5-fold cross-validation.

kernel is centered and normalized to have trace equal to one. We test the algorithms on several datasets taken from the UCI Machine Learning Repository (Asuncion & Newman, 2007) and Delve datasets (Rasmussen, 1996).

Table 3.1 summarizes our results. For classification, we compare against the `l1-svm` method and report the misclassification percentage. For regression, we compare against the `l2-krr` method and report RMSE. In general, we see that performance and alignment are well correlated. In all datasets, we see improvement over the uniform combination as well as the one-stage kernel learning algorithms. Note that although the `align` method often increases the alignment of the final kernel, as compared to the uniform combination, the `alignf` method gives the best alignment since it directly maximizes this quantity. Nonetheless, `align` provides an inexpensive heuristic that increases the alignment and performance of the final combination kernel.

To the best of our knowledge, these are the first kernel combination experiments for alignment with general base kernels. Previous experiments seem to have dealt exclusively with rank-1 base kernels built from the eigenvectors of a single kernel matrix (Cristianini et al., 2001). In the next section, we also examine rank-1 kernels, although not generated from a spectral decomposition.

Rank-1 kernel combinations

In this set of experiments we use the sentiment analysis dataset from Blitzer et al. (2007): *books*, *dvd*, *electronics* and *kitchen*. Each domain has 2,000 examples. In the regression setting, the goal is to predict a rating between 1 and 5, while for classification the goal is to discriminate positive (ratings ≥ 4) from negative reviews (ratings ≤ 2). We use rank-1 kernels based on the 4,000 most frequent bigrams. The kth base kernel, $\mathbf{K}_k$, corresponds to the k-th bigram count $\mathbf{v}_k$, $\mathbf{K}_k = \mathbf{v}_k \mathbf{v}_k^\top$. Each base kernel is normalized to have trace 1 and the labels are centered.

The `alignf` method returns a sparse weight vector due to the constraint $\boldsymbol{\mu} \geq \mathbf{0}$. As is demonstrated by the performance of the `l1-svm` method, Table 3.2, and also previously observed by Cortes et al. (2009a), a sparse weight vector $\boldsymbol{\mu}$ does not generally offer an improvement over the uniform combination in the rank-1 setting. Thus, we focus on the performance of `align` and compare it to `unif` and one-stage learning methods. Table 3.2 shows that `align` significantly improves both the alignment and the error percentage over `unif` and also improves somewhat over the one-stage `l2-krr` algorithm. Although the sparse weighting provided by `l1-svm` improves the alignment in certain cases, it does not improve performance.

Chapter 4

Conclusion

In this thesis, we have given *tight* margin-based bounds for several families of linearly combined kernels. We have shown bounds for convex combinations of kernels that exhibit only a *logarithmic* dependence on the number of base kernels. This encourages the use of a *very* large number of base kernels. Also, we have shown the first learning kernel type generalization bounds for the regression setting, using a specialized stability-based analysis unlike in previously shown bounds, and thereby extending previously studied scenarios.

On the algorithmic side, we designed a modified kernel ridge regression algorithm, LKRR, which uses a non-sparse combination of base kernels. This algorithm outperforms the uniform baseline and illustrates that previously suggested sparse combinations of kernels are not always beneficial. Finally, using a new alignment measure, we performed a series of experiments with two-stage algorithms where the kernel is selected separately from the learned hypothesis. The new alignment definition addresses an important problem that is illustrated with both artificial and real-world data. The two-stage experiments use general families of base kernels which, to the best of our knowledge, had not been investigated empirically before. In several settings, this method has shown improvement over the uniform baseline, as well as more complicated one-stage methods. We also presented a simpler and novel concentration bound that directly bounds the difference between the empirical and true measure of alignment. This shows that the alignment can be effectively estimated from samples. The algorithms presented in this thesis are made easily accessibly via the open-source OpenKernel library (Allauzen et al., 2010).

By using the presented algorithms, the burden on the user is lessened. Instead of being required to commit to a single kernel, the user is given the flexibility to define a general family of kernels by using a combination of multiple base kernels. Using a kernel that is automatically selected from such a family has shown better performance than a naive non-learned uniform combination kernel. These algorithmic results along with their theoretical foundations, have made it possible to include the problem of feature space selection within the data-driven framework of machine learning.

Although this thesis has made progress on several theoretical and algorithmic problems, there are also many remaining open problems. More complex families of non-linear combinations have been proposed (Varma & Babu, 2009; Cortes et al., 2009b), however, in several cases they will lead to non-convex optimization problems. In what ways can we restrict such problems in order to guarantee that we can efficiently find a reliable solution? What is the complexity of richer classes of kernels, such as polynomial, hierarchical or those induced by hyper-kernels, and what methods can be used to bound their complexities? In what situations do we expect such families to perform better than linear combinations?

Empirical results have shown improvements over the uniform combination of base kernels, but can even better performance be expected? Can multiple kernel algorithms compete with more complex baselines, such as kernels that are engineered by humans with the use of domain knowledge, e.g. sequence kernels used in biology domains (Ben-Hur & Noble, 2005; Allauzen et al., 2008)? How close are we to choosing the best kernel in hindsight from a given class of kernels?

In the future, it will also be interesting to make connections to the general scenario of automatic feature-space selection and more throughly understand what benefits and shortcomings multiple kernel learning algorithms exhibit in this broader setting. These are both important theoretical and practical questions, which will help drive the progress of automatic feature-space selection methods.

Appendix A

Supplementary Proofs

A.1 Lemmas for Stability-Based Bound

A.1.1 Expression of $\Delta\mu_k$

Lemma A.1. *For any samples S and S', $\Delta\mu_k$ can be expressed in terms of Δv_k as follows:*

$$\Delta\mu_k = \Lambda\left[\frac{\Delta v_k}{\|\mathbf{v}'\|} - \frac{v_k \sum_{i=1}^{p}(v_i + v_i')\Delta v_i}{\|\mathbf{v}\|\|\mathbf{v}'\|(\|\mathbf{v}\| + \|\mathbf{v}'\|)}\right]. \tag{A.1}$$

Proof. By definition of μ_k, we can write

$$\Delta\mu_k = \Lambda\left[\frac{v'_k}{\|\mathbf{v}'\|} - \frac{v_k}{\|\mathbf{v}\|}\right]$$

$$= \Lambda\left[\frac{v'_k - v_k}{\|\mathbf{v}'\|} - \frac{v_k\|\mathbf{v}'\| - v_k\|\mathbf{v}\|}{\|\mathbf{v}\|\|\mathbf{v}'\|}\right] = \Lambda\left[\frac{v'_k - v_k}{\|\mathbf{v}'\|} - \frac{v_k\Delta(\|\mathbf{v}\|)}{\|\mathbf{v}\|\|\mathbf{v}'\|}\right].$$

Observe that:

$$\Delta(\|\mathbf{v}\|) = \frac{\Delta(\|\mathbf{v}\|^2)}{\|\mathbf{v}\| + \|\mathbf{v}'\|} = \frac{\Delta(\sum_{i=1}^p v_i^2)}{\|\mathbf{v}\| + \|\mathbf{v}'\|} = \frac{\sum_{i=1}^p \Delta(v_i)(v_i + v'_i)}{\|\mathbf{v}\| + \|\mathbf{v}'\|}\ .$$

Plugging in this identity in the previous one yields the statement of the lemma.

□

A.1.2 Proof of Proposition 2.1

Proof. The terms $\Delta_K v_k$ appearing in V_1 have the following more explicit expression:

$$\Delta_K v_k = \Delta_K(\boldsymbol{\alpha}^\top \mathbf{K}_k(S')\boldsymbol{\alpha})$$

$$= \Delta_K(\boldsymbol{\alpha}^\top)\mathbf{K}_k(S')\boldsymbol{\alpha}' + \boldsymbol{\alpha}^\top \mathbf{K}_k(S')\Delta_K(\boldsymbol{\alpha}).$$

Thus, V_1 can be written as a sum $V_1 = V_{11} + V_{12}$ according to this decomposition. We shall show how V_{12} is bounded, V_{11} is bounded in a very similar way. In view of the expression for V_1 (2.27), and using $\mathbf{K}_k = \mathbf{\Phi}_k^\top \mathbf{\Phi}_k$, V_{12} can be written as

$$V_{12} = \Lambda \sum_{k=1}^{p} (\Delta_K \boldsymbol{\alpha})^\top \mathbf{Z} [\boldsymbol{\Phi}_k \boldsymbol{\alpha}]^\top \ , \tag{A.2}$$

with

$$\mathbf{Z} = \frac{\boldsymbol{\Phi}_k^\top \boldsymbol{\Phi}_k \boldsymbol{\alpha}}{\|\mathbf{v}\|} - \frac{v_k \sum_{i=1}^{p} \sum_i (v_i + v_i') \boldsymbol{\Phi}_i^\top \boldsymbol{\Phi}_i \boldsymbol{\alpha}}{\|\mathbf{v}\| \|\mathbf{v}'\| (\|\mathbf{v}\| + \|\mathbf{v}'\|)} \ . \tag{A.3}$$

Using the fact that $\|\boldsymbol{\Phi}_k \boldsymbol{\alpha}\|^2 = \boldsymbol{\alpha}^\top \boldsymbol{\Phi}_k^\top \boldsymbol{\Phi}_k \boldsymbol{\alpha} = \boldsymbol{\alpha}^\top \mathbf{K}_k \boldsymbol{\alpha} = v_k$ and similarly $\|\boldsymbol{\Phi}_i \boldsymbol{\alpha}\| = v_i^{1/2}$ and assuming without loss of generality that $\|v'\| \geq \|v\|$, V_{12} can be bounded as follows,

$$V_{12} \leq \Lambda \sum_{k=1}^{p} \|\Delta_K \boldsymbol{\alpha}\| \left(\frac{v_k}{\|\mathbf{v}\|} \|\boldsymbol{\Phi}_k\| + \frac{v_k}{\|\mathbf{v}\|} \frac{\sum_i (v_i + v_i') v_k^{1/2} v_i^{1/2} \|\boldsymbol{\Phi}_i\|}{\|\mathbf{v}'\| (\|\mathbf{v}\| + \|\mathbf{v}'\|)} \right) . \tag{A.4}$$

By the Cauchy-Schwarz inequality, the first sum $\sum_{k=1}^{p} \frac{v_k}{\|\mathbf{v}\|} \|\boldsymbol{\Phi}_k\|$ can be bounded as follows

$$\sum_{k=1}^{p} \frac{v_k}{\|\mathbf{v}\|} \|\boldsymbol{\Phi}_k\| \leq \frac{\|\mathbf{v}\|}{\|\mathbf{v}\|} \big(\sum_{k=1}^{p} \|\boldsymbol{\Phi}_k\|^2 \big)^{1/2} \leq R\sqrt{pm}, \tag{A.5}$$

since $\|\boldsymbol{\Phi}_k\| \leq R\sqrt{m}$. The second sum is similarly simplified and bounded as follows

$$\begin{aligned} &\sum_{k=1}^{p} \frac{v_k}{\|\mathbf{v}\|} \frac{\sum_{i=1}^{p} (v_i + v_i') v_k^{1/2} v_i^{1/2} \|\boldsymbol{\Phi}_i\|}{\|\mathbf{v}'\| (\|\mathbf{v}\| + \|\mathbf{v}'\|)} \\ &\qquad \leq \left(\sum_{k=1}^{p} \frac{v_k^{3/2}}{\|\mathbf{v}\|} \right) \left(\sum_{i=1}^{p} \frac{(v_i^{3/2} + v_i' v_i^{1/2})}{\|\mathbf{v}'\| (\|\mathbf{v}\| + \|\mathbf{v}'\|)} \right) \max_i \|\boldsymbol{\Phi}_i\|. \end{aligned}$$

In view of $\|\boldsymbol{\Phi}_i\| \leq R\sqrt{m}$ for all i, and using multiple applications of the Cauchy-Schwarz inequality, e.g., $\sum_{k=1}^{p} v_k^{3/2} = \sum_{k=1}^{p} v_k v_k^{1/2} \leq \|\mathbf{v}\| \|\mathbf{v}\|_1^{1/2}$ and

$\sum_{i=1}^p v_i' v_i^{1/2} \leq \|\mathbf{v}'\| \|\mathbf{v}\|_1^{1/2}$, the second sum is also bounded by $R\sqrt{pm}$ and $\|V_{12}\| \leq 2\Lambda R\sqrt{pm}\|\Delta_K \boldsymbol{\alpha}\|$. Proceeding in the same way for V_{11} leads to $\|V_{11}\| \leq 2\Lambda R\sqrt{pm}\|\Delta_K \boldsymbol{\alpha}\|$ and $\|V_1\| \leq 4\Lambda R\sqrt{pm}\|\Delta_K \boldsymbol{\alpha}\|$. □

A.1.3 Proof of Proposition 2.2

Proof. The main idea of the proof is to bound V_2 in terms of $\Delta_S \mathbf{w}$, the difference of the weight vectors h and h' already bounded in the proof of Theorem 2.2.

By definition, $v_k = \boldsymbol{\alpha}^\top \mathbf{K}_k \boldsymbol{\alpha}$. Since $\mathbf{K}_k = \boldsymbol{\Phi}_k^\top \boldsymbol{\Phi}_k$, then $v_k = \|\mathbf{w}_k\|^2$, where $\mathbf{w}_k = \boldsymbol{\Phi}_k(S)\boldsymbol{\alpha}$. Thus, in view of (2.27), V_2 can be written as follows

$$V_2 = \Lambda \sum_{k=1}^p \left(\frac{\Delta_S \|\mathbf{w}_k\|^2}{\|\mathbf{v}'\|} - \frac{v_k \sum_i (v_i + v_i') \Delta_S \|\mathbf{w}_i\|^2}{\|\mathbf{v}\|\|\mathbf{v}'\|(\|\mathbf{v}\| + \|\mathbf{v}'\|)} \right) \mathbf{w}_k^\top.$$

We can bound $|\Delta_S \|\mathbf{w}_k\|^2|$ in terms of $\|\Delta_S \mathbf{w}_k\|$:

$$\begin{aligned} |\Delta_S \|\mathbf{w}_k\|^2| &= |(\Delta_S \mathbf{w}_k)^\top \mathbf{w}_k' + \mathbf{w}_k^\top (\Delta_S \mathbf{w}_k)| \\ &= |(\Delta_S \mathbf{w}_k)^\top (\mathbf{w}_k' + \mathbf{w}_k)| \leq \|\mathbf{w}_k' + \mathbf{w}_k\| \|\Delta_S \mathbf{w}_k\|. \end{aligned}$$

Thus, since $\|\mathbf{w}_k\| = (\boldsymbol{\alpha}^\top \boldsymbol{\Phi}_k^\top \boldsymbol{\Phi}_k \boldsymbol{\alpha})^{1/2} \leq v_k^{1/2}$ and $\|\mathbf{w}_k'\| \leq {v'}_k^{1/2}$, $\|V_2\|$ can be bounded by

$$\|V_2\| \leq \Lambda\Bigg(\sum_{k=1}^{p} \frac{v_k^{1/2}(v_k^{1/2} + {v'}_k^{1/2})}{\|\mathbf{v}'\|} \|\Delta_S \mathbf{w}_k\| \\ + \sum_{i=1}^{p} \frac{(v_i + v_i')(v_i^{1/2} + {v'}_i^{1/2})}{\|\mathbf{v}\|\|\mathbf{v}'\|(\|\mathbf{v}\| + \|\mathbf{v}'\|)} \|\Delta_S \mathbf{w}_i\| \sum_{k=1}^{p} \|v_k \mathbf{w}_k^\top\| \Bigg).$$

The first sum can be bounded as follows

$$\sum_{k=1}^{p} \frac{v_k^{1/2}(v_k^{1/2} + {v'}_k^{1/2})\|\Delta_S \mathbf{w}_k\|}{\|\mathbf{v}'\|} \\ = \sum_{k=1}^{p} \frac{v_k + (v_k v_k')^{1/2}}{\mu_k \|\mathbf{v}'\|} \|\Delta_S(\mu_k \mathbf{w}_k)\| \\ \leq \Bigg(\underbrace{\Big(\sum_{k=1}^{p} \frac{(v_k + (v_k v_k')^{1/2})^2}{\mu_k^2 \|\mathbf{v}'\|^2} \Big)}_{F_1} \Big(\sum_{k=1}^{p} \|\Delta_S(\mu_k \mathbf{w}_k)\|^2 \Big) \Bigg)^{1/2}.$$

The first factor is bounded by a constant using multiple applications of the Cauchy-Schwarz inequality and assuming without loss of generality that $\|\mathbf{v}\| \leq \|\mathbf{v}'\|$:

$$F_1 = \sum_{k=1}^{p} \frac{v_k^2 + (v_k v_k') + 2 v_k^{3/2} {v'}_k^{1/2}}{\mu_k^2 \|\mathbf{v}'\|^2} \leq 4\,. \tag{A.6}$$

The second sum can be bounded as follows

$$
\begin{aligned}
&\frac{\sum_i (v_i + v_i')(v_i^{1/2} + v'^{1/2}_i)\|\Delta_S \mathbf{w}_i\|}{\|\mathbf{v}\|\|\mathbf{v}'\|(\|\mathbf{v}\| + \|\mathbf{v}'\|)} \sum_{k=1}^{p} \|v_k \mathbf{w}_k^\top\| \\
&\qquad \leq \sum_{i=1}^{p} \frac{(v_i + v_i')(v_i^{1/2} + v'^{1/2}_i)}{\mu_i \|\mathbf{v}\|\|\mathbf{v}'\|(\|\mathbf{v}\| + \|\mathbf{v}'\|)} \|\Delta_S(\mu_i \mathbf{w}_i)\| \sum_{k=1}^{p} \|v_k \mathbf{w}_k^\top\| \\
&\qquad \leq F_2 \left[\sum_{i=1}^{p} \|\Delta_S(\mu_i \mathbf{w}_i)\|^2 \right]^{1/2} \sum_{k=1}^{p} \|v_k \mathbf{w}_k^\top\|,
\end{aligned}
$$

where

$$
F_2 = \left[\sum_{i=1}^{p} \frac{(v_i + v_i')^2 (v_i^{1/2} + v'^{1/2}_i)^2}{\|\mathbf{v}\|^2 \|\mathbf{v}'\|^2 (\|\mathbf{v}\| + \|\mathbf{v}'\|)^2} \right]^{1/2} . \tag{A.7}
$$

The numerator of F_2, can be bounded using $\sum_{i=1}^{p} v_i^3 \leq \|\mathbf{v}\|^3$, $\sum_{i=1}^{p} v_i^{5/2} v'^{1/2}_i \leq \|\mathbf{v}\|^{5/2} \|\mathbf{v}'\|^{1/2}$ and applications of the Cauchy-Schwarz inequality such as

$$
\sum_{i=1}^{p} (v_i + v_i')^2 (v_i^{1/2} + v'^{1/2}_i)^2 \leq (\|\mathbf{v}\| + \|\mathbf{v}'\|)^2 (\|\mathbf{v}\|^{1/2} + \|\mathbf{v}'\|^{1/2})^2 .
$$

This leads to

$$
F_2 \leq \frac{\|\mathbf{v}\|^{1/2} + \|\mathbf{v}'\|^{1/2}}{\|\mathbf{v}\|\|\mathbf{v}'\|}
$$

and

$$
\|V_2\| \leq 2\Lambda \left(1 + \frac{\|\mathbf{v}\|^{1/2} + \|\mathbf{v}'\|^{1/2}}{2\|\mathbf{v}\|\|\mathbf{v}'\|} \sum_{k=1}^{p} \|v_k \mathbf{w}_k\| \right) F_3, \tag{A.8}
$$

with $F_3 = \left(\sum_{k=1}^{p} \|\Delta_S \mu_k \mathbf{w}_k\|^2 \right)^{1/2}$. If the feature vectors $\mathbf{w}_k$ are orthogonal, that is $\mathbf{w}_k^\top \mathbf{w}_{k'} = 0$ for $k \neq k'$ (which holds in particular if $\Phi_k(\mathbf{x}_i)^\top \Phi_{k'}(\mathbf{x}_i) = 0$

for $k \neq k'$ and $i \in \{1, \ldots, m\}$), then $F_3 = \|\Delta_S \mathbf{w}\|$ and

$$(\sum_{k=1}^{p} \|v_k \mathbf{w}_k\|)^2 = \| \sum_{k=1}^{p} v_k^2 \mathbf{w}_k^\top \mathbf{w}_k \|^2 = \sum_{k=1}^{p} v_k^3 \leq \|\mathbf{v}\|^3 .$$

Thus, using the bound on $\|\Delta_S \mathbf{w}\|$ from the proof of Theorem 2.2 yields

$$\begin{aligned} \|V_2\| &\leq 2\Lambda \left(1 + \frac{\|\mathbf{v}\|^{1/2} + \|\mathbf{v}'\|^{1/2}}{2\|\mathbf{v}\|\|\mathbf{v}'\|} \|\mathbf{v}\|^{3/2}\right) \|\Delta_S \mathbf{w}\| \\ &\leq 4\Lambda \|\Delta_S \mathbf{w}\| \leq \frac{4\Lambda M}{\lambda_{\min} + \lambda_0 m}. \end{aligned}$$ □

A.1.4 Proof of Proposition 2.3

Proof. Let $V = V_1 + V_2$ where V_1 (resp. V_2) is the expression corresponding to Δ_K (resp. Δ_S), where V and the terms corresponding to Δ_K and Δ_S are defined in equations (2.27), (2.28) and (2.29). We will denote by V_k, V_{1k} and V_{2k} each of the terms depending on k appearing in their sum.

The difference $\Delta_K \boldsymbol{\alpha} = -(\mathbf{K}' + \lambda \mathbf{I})^{-1} (\Delta_K \mathbf{K}) \boldsymbol{\alpha}$ can be expressed in terms of the V_ks as follows:

$$\Delta_K \boldsymbol{\alpha} = -(\mathbf{K}' + \lambda \mathbf{I})^{-1} \sum_{k=1}^{p} (V_k \boldsymbol{\Phi}_k)^\top.$$

Decomposing V_k as in $V_k = V_{1k} + V_{2k}$, using the expression of V_{1k} from (A.2), and collecting all $\Delta_K \boldsymbol{\alpha}$ terms to the left hand side, leads to the following expression relating $\Delta_K \boldsymbol{\alpha}$ to the V_{2k}s:

$$\Delta_K \boldsymbol{\alpha} = -\mathbf{Y}^{-1}\Big(\sum_{k=1}^{p}(V_{2k}\boldsymbol{\Phi}_k)^\top\Big), \tag{A.9}$$

with $\mathbf{Y} = \mathbf{K'} + \lambda\mathbf{I} + \Lambda\sum_{k=1}^{p}\frac{\mathbf{K}_k}{\|\mathbf{v}'\|}\boldsymbol{\alpha}\boldsymbol{\alpha}^\top\mathbf{Q}_k$, and $\mathbf{Q}_k = \big[\mathbf{K}_k - \frac{v_k}{\|\mathbf{v}\|}\frac{\sum_{i=1}^{p}(v_i+v_i')\mathbf{K}_i}{\|\mathbf{v}\|+\|\mathbf{v}'\|}\big]$. $\boldsymbol{\alpha}\boldsymbol{\alpha}^\top\mathbf{Q}_k$ has rank one since $\boldsymbol{\alpha}\boldsymbol{\alpha}^\top$ is a projection on the line spanned by $\boldsymbol{\alpha}$ and its trace $\mathrm{Tr}[\boldsymbol{\alpha}\boldsymbol{\alpha}^\top\mathbf{Q}_k] = \boldsymbol{\alpha}^\top\mathbf{Q}_k\boldsymbol{\alpha}$ is non-negative:

$$\begin{aligned}\boldsymbol{\alpha}^\top\mathbf{Q}_k\boldsymbol{\alpha} &= v_k - \frac{v_k}{\|\mathbf{v}\|}\frac{\sum_{i=1}^{p}(v_i^2 + v_i'v_i)}{\|\mathbf{v}\|+\|\mathbf{v}'\|}\\ &\geq v_k - \frac{v_k}{\|\mathbf{v}\|}\frac{\|\mathbf{v}\|^2 + \|\mathbf{v}'\|\|\mathbf{v}\|}{\|\mathbf{v}\|+\|\mathbf{v}'\|} = v_k - v_k = 0,\end{aligned}$$

using the Cauchy-Schwarz inequality. Thus, the eigenvalues of $\boldsymbol{\alpha}\boldsymbol{\alpha}^\top\mathbf{Q}_k$ are non-negative and since it has rank one and $\mathbf{K}_k$ is positive-semidefinite, the eigenvalues of $\mathbf{K}_k\boldsymbol{\alpha}\boldsymbol{\alpha}^\top\mathbf{Q}_k$ are also non-negative. This implies that the smallest eigenvalue of $\mathbf{Y}$ is at least λ and that $\|\mathbf{Y}^{-1}\| \leq 1/(\lambda_{\min} + \lambda_0 m)$. Finally, using the inequality $\|\sum_{k=1}^{p} V_{2k}\boldsymbol{\Phi}_k\| \leq \|V_2\|R\sqrt{m}$ completes the proof. □

A.2 Lemmas for Rademacher-Based Bounds

In the proof of Theorem 2.5, we need to upper bound the ratio $\binom{2r'}{2t_1,\ldots,2t_m}/\binom{r'}{t_1,\ldots,t_m}$. The following rough but straightforward inequality is sufficient to derive a bound on the Rademacher complexity in Theorem 2.5 with somewhat less favorable constants:

$$\begin{aligned}\tbinom{2r'}{2t_1,\ldots,2t_m} &= \frac{(2r')!}{(2t_1)!\cdots(2t_m)!} \le \frac{(2r')!}{(t_1)!\cdots(t_m)!}\\ &= \frac{(2r')\cdots(r'+1)\cdot r'!}{(t_1)!\cdots(t_m)!}\\ &\le \frac{(2r')^{r'}\cdot r'!}{(t_1)!\cdots(t_m)!} = (2r')^{r'}\tbinom{r'}{t_1,\ldots,t_m}.\end{aligned}$$

To further improve this result, the next lemma uses Stirling's approximation valid for all $n \ge 1$: $n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\lambda_n}$, with $\frac{1}{12n+1} < \lambda_n < \frac{1}{12n}$.

Lemma A.2. *For all $r' > 0$ and $t_1, \ldots, t_m$, it holds that:*

$$\tbinom{2r'}{2t_1,\ldots,2t_m} \le \left((1+\tfrac{1}{22})r'\right)^{r'}\tbinom{r'}{t_1,\ldots,t_m}.$$

Proof. By Stirling's formula,

$$\begin{aligned}\frac{(2r')!}{r'!} &= \sqrt{2}\left(\frac{2r'}{e}\right)^{2r'}\left(\frac{r'}{e}\right)^{-r'} e^{\lambda_{2r'}-\lambda_{r'}} \qquad \text{(A.10)}\\ &= \sqrt{2}\,2^{2r'}\left(\frac{r'}{e}\right)^{r'} e^{\lambda_{2r'}-\lambda_{r'}} = \sqrt{2}\left(\frac{4r'}{e}\right)^{r'} e^{\lambda_{2r'}-\lambda_{r'}}.\end{aligned}$$

Similarly, for any $t_i \ge 1$, we can write

$$\frac{t_i!}{(2t_i)!} = \frac{1}{\sqrt{2}}\left(\frac{e}{4t_i}\right)^{t_i} e^{\lambda_{t_i}-\lambda_{2t_i}} \le \frac{1}{\sqrt{2}}\left(\frac{e}{4}\right)^{t_i} e^{\lambda_{t_i}-\lambda_{2t_i}}.$$

Using $\sum_{i=1}^{m} t_i = \sum_{t_i \geq 1} t_i = r'$, we obtain:

$$\prod_{t_i \geq 1} \frac{t_i!}{(2t_i)!} \leq \frac{1}{\sqrt{2}} \Big(\frac{e}{4}\Big)^{r'} e^{\sum_{t_i \geq 1}(\lambda_{t_i} - \lambda_{2t_i})}. \tag{A.11}$$

In view of Eqn A.10 and A.11, the following inequality holds:

$$\frac{\binom{2r'}{2t_1,\ldots,2t_m}}{\binom{r'}{t_1,\ldots,t_m}} \leq (r')^{r'} e^{\lambda_{2r'} - \lambda_{r'} + \sum_{t_i \geq 1}(\lambda_{t_i} - \lambda_{2t_i})}.$$

We now derive an upper bound on the terms appearing in the exponent. Using the inequalities imposed on λ_{t_i} and λ_{2t_i} and the fact that the sum of t_is is r' leads to:

$$\begin{aligned}\sum_{t_i \geq 1} \lambda_{t_i} - \lambda_{2t_i} &\leq \sum_{t_i \geq 1} \frac{1}{12t_i} - \frac{1}{24t_i + 1} = \sum_{t_i \geq 1} \frac{12t_i + 1}{12t_i(24t_i + 1)} \\ &\leq \sum_{t_i \geq 1} \frac{1 + \frac{1}{12}}{24t_i + 1} \leq \sum_{t_i \geq 1} \frac{\frac{13}{12}}{25} \leq \frac{13r'}{300},\end{aligned}$$

and $\lambda_{2r'} - \lambda_{r'} \leq \frac{1}{24r'} - \frac{1}{12r'+1} \leq 0$. The inequality $e^{13/300} < 1 + 1/22$ then yields the statement of the lemma. □

A.3 Proof of Proposition 2.6

Proposition 2.6 *Let* $\mathbf{K}$ *and* $\mathbf{K}'$ *denote kernel matrices associated to the kernel functions* K *and* K' *for a sample of size* m *drawn according to* D*. Assume that for any* $x \in \mathcal{X}$, $K(x,x) \leq R^2$ *and* $K'(x,x) \leq R^2$*. Then, for any* $\delta > 0$, *with probability at least* $1-\delta$, *the following inequality holds:*

$$\left|\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2} - \mathrm{E}[K_c K'_c]\right| \leq \frac{18R^4}{m} + 24R^4\sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

The proof relies on a series of lemmas shown below.

Proof. By the triangle inequality and in view of Lemma A.5, the following holds:

$$\left|\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2} - \mathrm{E}[K_c K'_c]\right| \leq \left|\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2} - \mathrm{E}\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2}\right]\right| + \frac{18R^4}{m}.$$

Now, in view of Lemma A.4, the application of McDiarmid's inequality (McDiarmid, 1989) to $\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2}$ gives for any $\epsilon > 0$:

$$\Pr\left[\left|\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2} - \mathrm{E}\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2}\right]\right| > \epsilon\right] \leq 2\exp[-2m\epsilon^2/(24R^4)^2].$$

Setting δ to be equal to the right-hand side yields the statement of the proposition. □

We denote by $\mathbf{1} \in \mathbb{R}^{m\times 1}$ the vector with all entries equal to one, and by $\mathbf{I}$ the identity matrix.

Lemma A.3. *The following properties hold for centering kernel matrices:*

1. *For any kernel matrix* $\mathbf{K} \in \mathbb{R}^{m\times m}$, *the centered kernel matrix* $\mathbf{K}_c$ *can be given by*

$$\mathbf{K}_c = \left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]\mathbf{K}\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]. \tag{A.12}$$

2. For any two kernel matrices $\mathbf{K}$ *and* $\mathbf{K}'$*,*

$$\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F = \langle \mathbf{K}, \mathbf{K}'_c \rangle_F = \langle \mathbf{K}_c, \mathbf{K}' \rangle_F. \tag{A.13}$$

Proof. The first statement can be shown straightforwardly from the definition of $\mathbf{K}_c$ given by (2.39). The second statement follows from

$$\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F \quad = \quad \mathrm{Tr}\left[\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]\mathbf{K}\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]\mathbf{K}'\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]\right],$$

the fact that $[\mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^\top]^2 = \mathbf{I}_c = [\mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^\top]$, and the trace property $\mathrm{Tr}[\mathbf{AB}] = \mathrm{Tr}[\mathbf{BA}]$, valid for all matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times m}$. □

For a function f of the sample S, we denote by $\Delta(f)$ the difference $f(S') - f(S)$, where S' is a sample differing from S by just one point, say the m-th point is x_m in S and x'_m in S'. The following perturbation bound will be needed in order to apply McDiarmid's inequality.

Lemma A.4. *Let* $\mathbf{K}$ *and* $\mathbf{K}'$ *denote kernel matrices associated to the kernel functions* K *and* K' *for a sample of size* m *according to the distribution* D*. Assume that for any* $x \in \mathcal{X}$*,* $K(x,x) \leq R^2$ *and* $K'(x,x) \leq R^2$*. Then, the following perturbation inequality holds when changing one point of the sample:*

$$\frac{1}{m^2}|\Delta(\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F)| \leq \frac{24R^4}{m}.$$

Proof. By Lemma A.3, we can write:

$$\begin{aligned}
&\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F = \langle \mathbf{K}_c, \mathbf{K}' \rangle_F \\
&= \operatorname{Tr}\left[\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]\mathbf{K}\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]\mathbf{K}'\right] \\
&= \operatorname{Tr}\left[\mathbf{K}\mathbf{K}' - \frac{\mathbf{1}\mathbf{1}^\top}{m}\mathbf{K}\mathbf{K}' - \mathbf{K}\frac{\mathbf{1}\mathbf{1}^\top}{m}\mathbf{K}' + \frac{\mathbf{1}\mathbf{1}^\top}{m}\mathbf{K}\frac{\mathbf{1}\mathbf{1}^\top}{m}\mathbf{K}'\right] \\
&= \langle \mathbf{K}, \mathbf{K}' \rangle_F - \frac{\mathbf{1}^\top(\mathbf{K}\mathbf{K}' + \mathbf{K}'\mathbf{K})\mathbf{1}}{m} + \frac{(\mathbf{1}^\top\mathbf{K}\mathbf{1})(\mathbf{1}^\top\mathbf{K}'\mathbf{1})}{m^2}.
\end{aligned}$$

The perturbation of the first term is given by

$$\Delta(\langle \mathbf{K}, \mathbf{K}' \rangle_F) = \sum_{i=1}^{m} \Delta(\mathbf{K}_{im}\mathbf{K}'_{im}) + \Delta(\sum_{i \neq m} \mathbf{K}_{mi}\mathbf{K}'_{mi}).$$

By the Cauchy-Schwarz inequality, for any $i, j \in [1, m]$, $|\mathbf{K}_{ij}| = |K(x_i, x_j)| \leq \sqrt{K(x_i, x_i)K(x_j, x_j)} \leq R^2$. Thus,

$$\frac{1}{m^2}|\Delta(\langle \mathbf{K}, \mathbf{K}' \rangle_F)| \leq \frac{2m-1}{m^2}(2R^4) \leq \frac{4R^4}{m}.$$

Similarly, for the first part of the second term, we obtain

$$
\begin{aligned}
\frac{1}{m^2}\left|\Delta\left(\frac{\mathbf{1}^\top\mathbf{K}\mathbf{K}'\mathbf{1}}{m}\right)\right| &= \left|\Delta\left(\sum_{i,j,k=1}^{m}\frac{\mathbf{K}_{ik}\mathbf{K}'_{kj}}{m^3}\right)\right| \\
&= \left|\Delta\left(\frac{\sum_{i,k=1}^{m}\mathbf{K}_{ik}\mathbf{K}'_{km}+\sum_{i,j\neq m}\mathbf{K}_{im}\mathbf{K}'_{mj}}{m^3}\right.\right. \\
&\qquad \left.\left.+\frac{\sum_{k\neq m,j\neq m}\mathbf{K}_{mk}\mathbf{K}'_{kj}}{m^3}\right)\right| \\
&\leq \frac{m^2+m(m-1)+(m-1)^2}{m^3}(2R^4) \\
&\leq \frac{3m^2-3m+1}{m^3}(2R^4) \leq \frac{6R^4}{m}.
\end{aligned}
$$

Similarly, we have:

$$
\frac{1}{m^2}\left|\Delta\left(\frac{\mathbf{1}^\top\mathbf{K}'\mathbf{K}\mathbf{1}}{m}\right)\right| \leq \frac{6R^4}{m}. \tag{A.14}
$$

The final term is bounded as follows,

$$
\begin{aligned}
\frac{1}{m^2}\left|\Delta\left(\frac{(\mathbf{1}^\top\mathbf{K}\mathbf{1})(\mathbf{1}^\top\mathbf{K}'\mathbf{1})}{m^2}\right)\right| &\leq \left|\Delta\left(\frac{\sum_{i,j,k}\mathbf{K}_{ij}\mathbf{K}'_{km}+\sum_{i,j,k\neq m}\mathbf{K}_{ij}\mathbf{K}'_{mk}}{m^4}\right.\right. + \\
&\qquad \left.\left.\frac{\sum_{i,j\neq m,k\neq m}\mathbf{K}_{im}\mathbf{K}'_{jk}+\sum_{i\neq m,j\neq m,k\neq m}\mathbf{K}_{mi}\mathbf{K}'_{jk}}{m^4}\right)\right| \\
&\leq \frac{m^3+m^2(m-1)+m(m-1)^2+(m-1)^3}{m^4}(2R^4) \\
&\leq \frac{8R^4}{m}.
\end{aligned}
$$

Combining these last four inequalities leads directly to the statement of the lemma. □

Because of the diagonal terms of the matrices, $\frac{1}{m^2}\langle\mathbf{K}_c,\mathbf{K}'_c\rangle_F$ is not an unbiased estimate of $\mathrm{E}[K_c K'_c]$. However, as shown by the following lemma, the estimation bias decreases at the rate $O(1/m)$.

Lemma A.5. *Under the same assumptions as Lemma A.4, the following bound on the difference of expectations holds:*

$$\left| \mathop{\mathrm{E}}_{x,x'}[K_c(x,x')K'_c(x,x')] - \mathop{\mathrm{E}}_{S}\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2}\right] \right| \leq \frac{18R^4}{m}.$$

Proof. To simplify the notation, unless otherwise specified, the expectation is taken over x, x' drawn according to the distribution D.

The key observation used in this proof is that

$$\mathop{\mathrm{E}}_{S}[\mathbf{K}_{ij}\mathbf{K}'_{ij}] = \mathop{\mathrm{E}}_{S}[K(x_i,x_j)K'(x_i,x_j)] = \mathrm{E}[KK'], \tag{A.15}$$

for i, j distinct. For expressions such as $\mathrm{E}_S[\mathbf{K}_{ik}\mathbf{K}'_{kj}]$ with i, j, k distinct, we obtain the following:

$$\mathop{\mathrm{E}}_{S}[\mathbf{K}_{ik}\mathbf{K}'_{kj}] = \mathop{\mathrm{E}}_{S}[K(x_i,x_k)K'(x_k,x_j)] = \mathop{\mathrm{E}}_{x'}[\mathop{\mathrm{E}}_{x}[K]\mathop{\mathrm{E}}_{x}[K']]. \tag{A.16}$$

Let us start with the expression of $\mathrm{E}[K_c K'_c]$:

$$\begin{aligned}\mathrm{E}[K_cK'_c] = \mathrm{E}\Big[&\big(K - \mathop{\mathrm{E}}_{x'}[K] - \mathop{\mathrm{E}}_{x}[K] + \mathrm{E}[K]\big)\\ &\qquad\big(K' - \mathop{\mathrm{E}}_{x'}[K'] - \mathop{\mathrm{E}}_{x}[K'] + \mathrm{E}[K']\big)\Big].\end{aligned} \tag{A.17}$$

After expanding this expression, applying the expectation to each of the terms, and simplifying, we obtain:

$$\mathrm{E}[K_c K_c'] = \mathrm{E}[KK'] - 2 \mathop{\mathrm{E}}_{x} \big[\mathop{\mathrm{E}}_{x'}[K] \mathop{\mathrm{E}}_{x'}[K'] \big] + \mathrm{E}[K]\,\mathrm{E}[K'].$$

$\langle \mathbf{K}_c, \mathbf{K}_c' \rangle_F$ can be expanded and written more explicitly as follows:

$$\begin{aligned}
&\langle \mathbf{K}_c, \mathbf{K}_c' \rangle_F \\
&= \langle \mathbf{K}, \mathbf{K}' \rangle_F - \frac{\mathbf{1}^\top \mathbf{K}\mathbf{K}'\mathbf{1}}{m} - \frac{\mathbf{1}^\top \mathbf{K}'\mathbf{K}\mathbf{1}}{m} + \frac{\mathbf{1}^\top \mathbf{K}'\mathbf{1}\mathbf{1}^\top \mathbf{K}\mathbf{1}}{m^2} \\
&= \sum_{i,j=1}^{m} \mathbf{K}_{ij}\mathbf{K}'_{ij} - \frac{1}{m} \sum_{i,j,k=1}^{m} (\mathbf{K}_{ik}\mathbf{K}'_{kj} + \mathbf{K}'_{ik}\mathbf{K}_{kj}) + \\
&\quad \frac{1}{m^2} (\sum_{i,j=1}^{m} \mathbf{K}_{ij})(\sum_{i,j=1}^{m} \mathbf{K}'_{ij}).
\end{aligned}$$

To take the expectation of this expression, we shall use the observations (A.15) and (A.16) and similar identities. Counting terms of each kind, leads to the following expression of the expectation:

$$
\begin{aligned}
\mathop{\mathrm{E}}_{S}\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2}\right] &= \left[\frac{m(m-1)}{m^2} - \frac{2m(m-1)}{m^3} + \frac{2m(m-1)}{m^4}\right] \mathrm{E}[KK'] \\
&+ \left[\frac{-2m(m-1)(m-2)}{m^3} + \frac{2m(m-1)(m-2)}{m^4}\right] \\
&\quad \mathop{\mathrm{E}}_{x}\Big[\mathop{\mathrm{E}}_{x'}[K] \mathop{\mathrm{E}}_{x'}[K']\Big] \\
&+ \left[\frac{m(m-1)(m-2)(m-3)}{m^4}\right] \mathrm{E}[K]\,\mathrm{E}[K'] \\
&+ \left[\frac{m}{m^2} - \frac{2m}{m^3} + \frac{m}{m^4}\right] \mathop{\mathrm{E}}_{x}[K(x,x)K'(x,x)] \\
&+ \left[\frac{-m(m-1)}{m^3} + \frac{2m(m-1)}{m^4}\right] \mathrm{E}[K(x,x)K'(x,x')] \\
&+ \left[\frac{-m(m-1)}{m^3} + \frac{2m(m-1)}{m^4}\right] \mathrm{E}[K(x,x')K'(x,x)] \\
&+ \left[\frac{m(m-1)}{m^4}\right] \mathop{\mathrm{E}}_{x}[K(x,x)] \mathop{\mathrm{E}}_{x}[K'(x,x)] \\
&+ \left[\frac{m(m-1)(m-2)}{m^4}\right] \mathop{\mathrm{E}}_{x}[K(x,x)]\,\mathrm{E}[K'] \\
&+ \left[\frac{m(m-1)(m-2)}{m^4}\right] \mathrm{E}[K] \mathop{\mathrm{E}}_{x}[K'(x,x)].
\end{aligned}
$$

Taking the difference with the expression of $\mathrm{E}[K_c K'_c]$ (Equation A.17), using the fact that terms of form $\mathrm{E}_x[K(x,x)K'(x,x)]$ and other similar ones are all bounded by R^4 and collecting the terms gives

$$\begin{aligned}
&\left|\operatorname{E}[K_c K_c'] - \operatorname*{E}_S\left[\frac{\langle \mathbf{K}_c, \mathbf{K}_c' \rangle_F}{m^2}\right]\right| \leq \frac{3m^2-4m+2}{m^3}\operatorname{E}[KK'] \\
&- 2\frac{4m^2-5m+2}{m^3}\operatorname*{E}_x\left[\operatorname*{E}_{x'}[K]\operatorname*{E}_{x'}[K']\right] \\
&+ \frac{6m^2-11m+6}{m^3}\operatorname{E}[K]\operatorname{E}[K'] + \gamma,
\end{aligned}$$

with $|\gamma| \leq \frac{m-1}{m^2}R^4$. Using again the fact that the expectations are bounded by R^4 yields

$$\begin{aligned}
\left|\operatorname{E}[K_c K_c'] - \operatorname*{E}_S\left[\frac{\langle \mathbf{K}_c, \mathbf{K}_c' \rangle_F}{m^2}\right]\right| &\leq \left[\frac{3}{m} + \frac{8}{m} + \frac{6}{m} + \frac{1}{m}\right] R^4 \\
&\leq \frac{18}{m}R^4,
\end{aligned}$$

and concludes the proof. □

Bibliography

Allauzen, C., Mohri, M., & Rostamizadeh, A. (2010). Openkernel. `www.openkernel.org`.

Allauzen, C., Mohri, M., & Talwalkar, A. (2008). Sequence kernels for predicting protein essentiality. *Proceedings of the 25th International Conference on Machine Learning*.

Anthony, M., & Bartlett, P. L. (1999). *Neural network learning: Theoretical foundations*. Cambridge University Press.

Argyriou, A., Micchelli, C., & Pontil, M. (2005). Learning convex combinations of continuously parameterized basic kernels. *Proceedings of the 18th Annual Conference on Learning Theory*.

Asuncion, A., & Newman, D. (2007). UCI machine learning repository. `http://archive.ics.uci.edu/ml/`.

Bach, F. (2008). Exploring large feature spaces with hierarchical multiple kernel learning. *Advances in Neural Information Processing Systems 21*.

Bach, F., Lanckriet, G., & Jordan, M. (2004). Multiple kernel learning, conic duality, and the SMO algorithm. *Proceedings of the 21st International Conference on Machine Learning.*

Balcan, M.-F., & Blum, A. (2006). On a theory of learning with similarity functions. *Proceedings of the 23rd International Conference on Machine Learning.*

Bartlett, P. L., & Mendelson, S. (2002). Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, *3*.

Baudat, G., & Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Computation*, *12*.

Ben-Hur, A., & Noble, W. (2005). Kernel methods for predicting protein-protein interactions. *Bioinformatics*, *21*.

Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, Bollywood, Boomboxes and Blenders: Domain Adaptation for Sentiment Classification. *Association for Computational Linguistics.*

Blum, A., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, *97*.

Boser, B., Guyon, I., & Vapnik, V. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the 5th Annual Conference on Learning Theory.*

Bousquet, O., & Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research*, *2*.

Bousquet, O., & Herrmann, D. J. L. (2002). On the complexity of learning the kernel matrix. *Advances in Neural Information Processing Systems 15*.

Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*, *46*.

Cortes, C. (2009). Invited talk: Can learning kernels help performance? *Proceedings of the 26th International Conference on Machine Learning*.

Cortes, C., Gretton, A., Lanckriet, G., Mohri, M., & Rostamizadeh, A. (2008a). Kernel learning: Automatic selection of optimal kenrels. *Neural Information Processing Systems: Workshop*.

Cortes, C., Haffner, P., & Mohri, M. (2004). Rational Kernels: Theory and Algorithms. *Journal of Machine Learning Research*, *5*.

Cortes, C., Mohri, M., & Rostamizadeh, A. (2008b). Learning sequence kernels. *IEEE Workshop on Machine Learning for Signal Processing, 2008*.

Cortes, C., Mohri, M., & Rostamizadeh, A. (2009a). L_2 regularization for learning kernels. *Proceedings of the 25th Annual Conference on Uncertainty in Artificial Intelligence*.

Cortes, C., Mohri, M., & Rostamizadeh, A. (2009b). Learning non-linear combinations of kernels. *Advances in Neural Information Processing Systems 22*.

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning, 20*.

Cristianini, N., Campbell, C., & Shawe-Taylor, J. (1999). Dynamically adapting kernels in support vector machines. *Advances in Neural Information Processing Systems 12*.

Cristianini, N., Kandola, J. S., Elisseeff, A., & Shawe-Taylor, J. (2002). On kernel target alignment. http://www.support-vector.net/papers/alignment_JMLR.ps, unpublished.

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., & Kandola, J. S. (2001). On kernel-target alignment. *Advances in Neural Information Processing Systems 14*.

Devroye, L., Györfi, L., & Lugosi, G. (1996). *A probabilistic theory of pattern recognition*. Springer.

Dhillon, I., Guan, Y., & Kulis, B. (2004). Kernel k-means: spectral clustering and normalized cuts. *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Drucker, H., Burges, C., Kaufman, L., Smola, A., & Vapnik, V. (1997). Support vector regression machines. *Advances in Neural Information Processing Systems 10*.

Grandvalet, Y., & Canu, S. (2003). Adaptive scaling for feature selection in SVMs. *Advances in Neural Information Processing Systems 16*.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research*, *3*.

Kakade, S. M., Sridharan, K., & Tewari, A. (2009). On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. *Advances in Neural Information Processing Systems 22*.

Kandola, J. S., Shawe-Taylor, J., & Cristianini, N. (2002a). *On the extensions of kernel alignment* (Technical Report 120). Department of Computer Science, University of London, UK.

Kandola, J. S., Shawe-Taylor, J., & Cristianini, N. (2002b). *Optimizing kernel alignment over combinations of kernels* (Technical Report 121). Department of Computer Science, University of London, UK.

Kloft, M., Brefeld, U., Sonnenburg, S., Laskov, P., Müller, K., & Zien, A. (2009). Efficient and accurate lp-norm multiple kernel learning. *Advances in Neural Information Processing Systems 19*.

Kohavi, R., & John, G. (1997). Wrappers for feature subset selection. *Artificial intelligence*, *97*.

Koltchinskii, V., & Panchenko, D. (2002). Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, *30*.

Koltchinskii, V., & Yuan, M. (2008). Sparse recovery in large ensembles of kernel machines on-line learning and bandits. *Proceedings of the 21st Annual Conference on Learning Theory*.

Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L. E., & Jordan, M. (2002). Learning the kernel matrix with semidefinite programming. *Proceedings of the 19th International Conference on Machine Learning*.

Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L. E., & Jordan, M. (2004a). Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, *5*.

Lanckriet, G., Deng, M., Cristianini, N., Jordan, M., & Noble, W. (2004b). Kernel-based data fusion and its application to protein function prediction in yeast. *Pacific Symposium on Biocomputing*.

Lecun, Y., & Cortes, C. (1998). The mnist database of handwritten digits. `http://http://yann.lecun.com/exdb/mnist/`.

Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge and data engineering*.

McDiarmid, C. (1989). On the method of bounded differences. *Surveys in combinatorics*, *141*.

Meila, M. (2003). Data centering in feature space. *9th International Workshop on Artificial Intelligence and Statistics*.

Micchelli, C., & Pontil, M. (2005). Learning the kernel function via regularization. *Journal of Machine Learning Research*, *6*.

Ong, C. S., Smola, A., & Williamson, R. (2005). Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, *6*.

Pothin, J.-B., & Richard, C. (2008). Optimizing kernel alignment by data translation in feature space. *International Conference on Acoustics, Speach and Signal Processing*.

Rakotomamonjy, A., Bach, F., Grandvalet, Y., & Canu, S. (2008). SimpleMKL. *Journal of Machine Learning Research*, *9*.

Rasmussen, C. E. (1996). Delve datasets. `http://www.cs.toronto.edu/~delve/data/datasets.html`.

Saunders, C., Gammerman, A., & Vovk, V. (1998). Ridge Regression Learning Algorithm in Dual Variables. *Proceedings of the 15th International Conference on Machine Learning*.

Schölkopf, B., & Smola, A. (2002). *Learning with kernels*. MIT Press: Cambridge, MA.

Schölkopf, B., & Smola, A. (2003). A short introduction to learning with kernels. *Advanced lectures on machine learning*.

Schölkopf, B., Smola, A., & Müller, K. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*.

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

Sonnenburg, S., Ratsch, G., & Schafer, C. (2006). A general and efficient multiple kernel learning algorithm. *Advances in Neural Information Processing Systems 19*.

Srebro, N., & Ben-David, S. (2006). Learning bounds for support vector machines with learned kernels. *Proceedings of the 19th Annual Conference on Learning Theory*.

Vapnik, V., & Chervonenkis, A. (1974). *Theory of pattern recognition*. Nauka, Moscow.

Vapnik, V. N. (1998). *Statistical learning theory*. John Wiley & Sons.

Varma, M., & Babu, B. R. (2009). More generality in efficient multiple kernel learning. *Proceedings of the 26th International Conference on Machine Learning*.

von Neumann, J. (1937). Uber ein ökonomisches Gleichungssystem. *Ergebnisse Mathematischen Kolloquiums*.

Wahba, G., Gu, C., & Wang, Y. (1993). Soft classification, aka risk estimation, via penalized log likelihood and smoothing spline analysis of variance. *The Mathematics of Generalization-The Proceedings of the SFI/CNLS Workshop on Formal Approaches to Supervised Learning*.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2001). Feature selection for SVMs. *Advances in Neural Information Processing Systems 14*.

Xu, L., Neufeld, J., Larson, B., & Schuurmans, D. (2005). Maximum margin clustering. *Advances in Neural Information Processing Systems 18*.

Xu, Z., Jin, R., King, I., & Lyu, M. (2009). An extended level method for efficient multiple kernel learning. *Advances in Neural Information Processing Systems 22*.

Ying, Y., & Campbell, C. (2009). Generalization bounds for learning the kernel problem. *Proceedings of the 22nd Annual Conference on Learning Theory.*