Kernels: Regularization and Optimization

Cheng Soon Ong

A thesis submitted for the degree of
Doctor of Philosophy at
The Australian National University

April 2005

Typeset in Computer Modern by TEX and LATEX 2_ε.

Except where otherwise indicated, this thesis is my own original work.

Cheng Soon Ong
12 April 2005

To my parents,
Ong Chee Lai and Ooi Eng Wah.

Acknowledgements

There are many people who have directly or indirectly made it possible for me to continue on this journey of discovery. At the risk of forgetting to mention some, I would like to sincerely thank the following people (in no particular order): The Australian Taxpayer, Michelle Moravec, Pamela Philips, Eric McCreath, Zoubin Ghahramani, Ulrike von Luxburg, Matthias Hein, Agnes Boskovitz, Edward Harrington, Evan Greensmith, Doug Aberdeen, S.V.N. Vishwanathan, Lydia Knüfing, Markus Hegland, Lai Weng Kin, John Shawe-Taylor and Bernhard Schölkopf.

I would especially like to thank my supervisors John W. Lloyd and Alexander J. Smola for all the support and advice that they have given me these past few years. They have made sure that I did not get too distracted by the subtle hedonistic pleasures of Canberra, and paved the way for me to have very productive research visits to the Gatsby Institute in University College London, Microsoft Research Cambridge, the Max Planck Institute in Tübingen and the University of Southampton. Stéphane Canu, whom I shared an office with for several highly productive months when he visited ANU, had a large influence on the later chapters of this thesis. I would also like to thank the co-authors of my conference publications and submissions Xavier Mary, Petra Philips and Bob Williamson for the pleasure for working with them and for their permission to use our joint work in this thesis. Large parts of Chapter 2 was published in Ong et al. [2003], and a shortened version of Chapter 3 was published in Ong and Smola [2003]. The main ideas of Chapter 4 was published in Ong et al. [2004]. Although I was the principal author on the above publications all the work was done jointly with my coauthors and it is impossible to disentangle each individual contribution.

I am indebted to the Australian National University Malaysian Alumni scholarship for supporting me financially during the period of my PhD, and Mimos Berhad who kindly allowed me to be on study leave all this time, hence allowing me to do my research without worrying about where my next meal will come from.

Last but not least, I thank my friends and family who have kept me sane and reminded me that there is more to life than a PhD.

Abstract

This thesis extends the paradigm of machine learning with kernels. This paradigm is based on the idea of generalizing an inner product between vectors to a similarity measure between objects. The kernel implicitly defines a feature mapping between the space of objects and the space of functions, called the reproducing kernel Hilbert space. There have been many successful applications of positive semidefinite kernels in diverse fields. Among the reasons for its success are a theoretically motivated regularization method and efficient algorithms for optimizing the resulting problems.

Since the kernel has to effectively capture the domain knowledge in an application, we study the problem of learning the kernel itself from training data. The proposed solution is a kernel on the space of kernels itself, which we called a hyperkernel. This provides a method for regularization via the norm of the kernel. We show that for several machine learning tasks, such as binary classification, regression and novelty detection, the resulting optimization problem is a semidefinite program. We solve the corresponding optimization problems using the same parameter settings across all problems, and demonstrate that we have further automated machine learning methods.

We observe that the restriction for kernels to be positive semidefinite can be removed. The non-positive kernels, called indefinite kernels, have corresponding functional theory, and define reproducing kernel Kreĭn spaces. We derive machine learning problems with indefinite kernels and prove the representer theorem as well as generalization error bounds.

We provide theoretical and experimental evidence to support the idea of regularization by early stopping of conjugate gradient type algorithms. Conjugate gradient type algorithms are iterative methods that generate solutions in Krylov subspaces, and exhibit semi-convergence. We analyse the sequence of Krylov subspaces that determine the associated filter function on the spectrum of the inverse problem, and quantitatively investigate semi-convergence. These algorithms are then used for machine learning with indefinite kernels.

Contents

List of Symbols

These are the symbols used in the thesis, and the page where they are first defined

K	Gram Matrix of kernel function, $K_{ij} := k(x_i, x_j)$ where $x_i, x_j \in \mathcal{X}$, page 14
R	Expected Risk, page 5
$R_m(\mathcal{F})$	Rademacher average of a set of functions $\mathcal{F}$, page 60
T	evaluation functional, page 52
$\overline{\mathcal{K}}$	Hilbert space associated with Kreĭn space, page 51
$\mathbb{R}$	the real numbers, page 3
Z	Set of training data and labels, page 25
$\mathcal{X}$	Space of input data, page 2
X_{train}	Training data, page 2
$\det K$	determinant of the matrix K, page 14
Q_{emp}	Empirical Quality Functional, page 15
R_{emp}	Empirical Risk Functional, page 15
$\underline{k}(x, y, s, t)$	Hyperkernel, page 21
$\mathcal{S}_k(z; G)$	Krylov subspace of rank k, page 70
$\mathcal{Y}$	Space of input labels, page 2
Y_{train}, y	Training labels, page 2
λ_i	ith eigenvalue, page 61
ℓ	Loss function, page 5
Q	Expected Quality Functional, page 15
R	Expected Risk Functional, page 15

Q_{reg}	Regularised Quality Functional, page 21
$\mathcal{K}$	reproducing kernel Kreĭn space, page 52
σ_i	ith singular value, page 64
spec$A$	spectrum of the operator A, page 62
tr$K$	trace of the matrix K, page 14
$k(x_i, x_j)$	kernel, page 3
$p_k(A)$	residual polynomials of order k, page 71
$q_k(A)$	iteration polynomials of order k, page 71

Chapter 1

Introduction

This chapter introduces kernels, regularization and optimization, and shows their role in machine learning. These are the key concepts we deal with when extending the framework of machine learning with kernels. It concludes with a description of the contributions of this thesis.

1.1 Introduction to Machine Learning

Machine Learning is an attempt at codifying a particularly useful human trait, which is the ability to generalise previous experiences to make educated guesses about the future behaviour of the world. In practice, a computer can only make observations of a phenomenon. The task of a machine learning algorithm is to use the observations and try to predict future observations. The algorithm uses the examples to form a hypothesis about the underlying phenomenon. The algorithms we investigate in this thesis are batch algorithms, which means that there are distinct training and testing phases. Hence we collect a set of observations, which are also called training examples, and update the hypothesis based on the examples. This is called the training phase. Once the machine is trained, the resulting hypothesis is used to perform predictions.

Since we cannot take perfect measurements of the world, some of the training data may be misleading. One can view this as noise in the data. To consistently and efficiently predict the behaviour of a real world problem, we effectively have to restrict the class of hypotheses about the world. One popular restriction uses the kernel of an integral operator, and is called kernel methods [Schölkopf and Smola, 2002, Shawe-Taylor and Cristianini, 2004]. This thesis investigates kernel methods for machine learning. For the moment, we treat the kernel machine as a black box and introduce the other parts of machine learning in more detail.

We assume that we can learn the nature of the underlying problem by investigating a sample of instances from the problem. Figure 1.1 depicts the framework of learning the kernel as described below. In many applications, each observation of the training

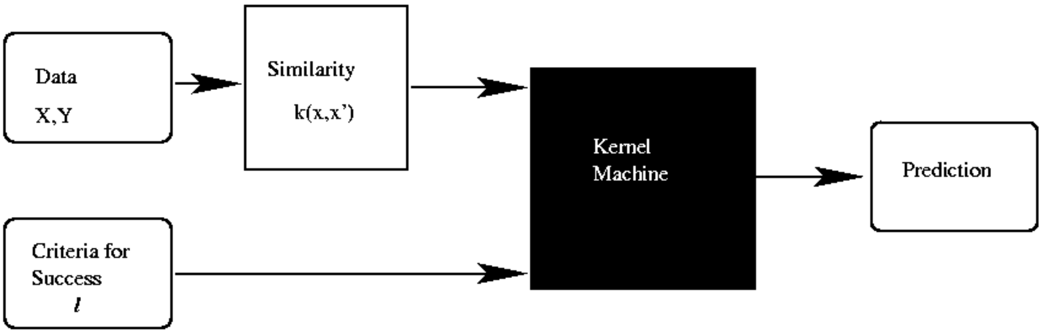


Figure 1.1: A representation of the learning with kernels framework. The data and criteria for success are determined by the problem. Prior knowledge about problem is captured by the similarity measure. The kernel machine produces a prediction for future examples of data.

data is labelled, for example by some human expert. This is called supervised learning. We denote by $\mathcal{X}$ the space of input data and $\mathcal{Y}$ the space of labels (if we have a supervised learning problem). Denote by $X_{\text{train}} := \{x_1, \ldots, x_m\}$ the training data and with $Y_{\text{train}} := \{y_1, \ldots, y_m\}$ a set of corresponding labels, jointly drawn independently and identically from some probability distribution $\Pr(x, y)$ on $\mathcal{X} \times \mathcal{Y}$. We shall, by convenient abuse of notation, generally denote Y_{train} by the vector y when writing equations in matrix notation. For a new example $x_{\text{test}} \in \mathcal{X}$, the problem of machine learning is to predict the label y_{test} using our prior knowledge of the problem and the training examples. Observe that we do not know $\Pr(x, y)$, and hence the algorithm has to perform predictions based on the information provided by training data. The success of our algorithm is measured via the loss function ℓ which is a function of the training data, and is described in more detail in Section 1.1.2.

The class of algorithms we consider do not deal directly with the objects themselves ($x_i \in \mathcal{X}$), but with measures of similarity between them. In other words, our algorithms only use $k(x_i, x_j)$, where k is a similarity measure between any two observations. Such algorithms are described in Section 1.1.3. The assumption is that the similarity captures the domain knowledge. Furthermore, the hypothesis class contains only linear hypotheses. This restriction is not as severe as it may seem, as we shall see in the next section. The advantage of this approach is that we can separate the design of good algorithms for machine learning from the design of good representations of the data based on domain knowledge.

1.1.1 Kernels

Kernels are a special type of similarity measure with many "nice" properties. Kernels are inner products, denoted by $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, in a feature space $\mathcal{H}$. That is, for some $x_i, x_j \in \mathcal{X}$,

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}},$$

where $\phi : \mathcal{X} \to \mathcal{H}$ is a mapping from the space of objects to a feature space, and the inner product is with respect to a Hilbert space $\mathcal{H}$, with kernel $k(\cdot, \cdot)$ [Aronszajn, 1950]. In this setting, the hypothesis of the machine learning algorithm is a function $f : \mathcal{X} \to \mathbb{R}$ in the feature space defined by ϕ. Kernel methods do not explicitly use the feature mapping, instead they implicitly define ϕ via the kernel function k. A linear machine learning algorithm can then be applied to these features. However, since ϕ can be a nonlinear mapping, the algorithm can estimate nonlinear functions as well.

Given our hypothesis function f, we apply it at the test point x_{test}, that is we compute $f(x_{\text{test}})$. Formally this is called evaluating the function at a given point. In addition, since what we really want is to predict well, we would like functions which are similar to each other to also give predictions which are similar to each other. This means that if we learn a function which is close to a good function, it will predict labels close to the correct labels. In functional analysis this property is called continuity of the evaluation functional. A reproducing kernel Hilbert space (RKHS) consists of functions which have these properties of being pointwise defined and have a continuous evaluation functional. There are two possible ways to define a RKHS, giving alternative views. The first definition tells us about the restrictions we place on our functions and their evaluations.

Definition 1 (Reproducing Kernel Hilbert Space) *[Schwartz, 1964] A reproducing kernel Hilbert space is a space $\mathcal{H}$ for which at each $x \in \mathcal{X}$ the Dirac evaluation functional,*

$$\delta_x : \mathcal{H} \to \mathbb{R},$$

which maps $f \in \mathcal{H}$ to $f(x) \in \mathbb{R}$, is a bounded (or equivalently, continuous) linear functional.

From this definition, and using the Riesz representation theorem [Akhiezer and Glazman, 1993, Section 16], we get an alternative but equivalent construction that views the space in terms of its reproducing property. This second definition appears more frequently in machine learning literature.

Definition 2 (Reproducing Kernel Hilbert Space) *[Aronszajn, 1950] Let $\mathcal{X}$ be a nonempty set (the index set) and denote by $\mathcal{H}$ a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$.*

$\mathcal{H}$ is called a reproducing kernel Hilbert space endowed with the dot product $\langle\cdot,\cdot\rangle$ (and the norm $\|f\| := \sqrt{\langle f, f\rangle}$) if there exists a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with the following properties.

1. *k has the reproducing property*

$$\langle f, k(x,\cdot)\rangle = f(x) \text{ for all } f \in \mathcal{H}, x \in \mathcal{X}; \tag{1.1}$$

   in particular, $\langle k(x,\cdot), k(x',\cdot)\rangle = k(x,x')$ for all $x, x' \in \mathcal{X}$.

2. *k spans $\mathcal{H}$, i.e. $\mathcal{H} = \overline{\operatorname{span}\{k(x,\cdot)|x \in \mathcal{X}\}}$ where $\overline{X}$ is the completion of the pre-Hilbert space X.*

The symmetric function of two arguments, $k(x_i, x_j)$, is called the reproducing kernel. This function is positive semidefinite and has been called a Mercer kernel or a positive semidefinite kernel.

Definition 3 (Positive Semidefinite Kernel) *A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a* positive semidefinite kernel *if it is positive semidefinite, that is, for all $a_1, \ldots, a_n \in \mathbb{R}$ and $x_1, \ldots, x_n \in \mathcal{X}$,*

$$\sum_{i,j=1}^{n} a_i a_j k(x_i, x_j) \geqslant 0.$$

In many cases, when it is clear from the context, we refer to the positive semidefinite kernel as just the kernel, or the positive kernel. In the later chapters, where we extend this notion of a kernel, we note when the kernel is not positive semidefinite.

A key question is whether this relationship between the kernel and the RKHS is unique. If for each kernel function $k(\cdot,\cdot)$, there were many possible RKHSs, we would not be able to specify precisely which class of functions we are using. Similarly, difficulties would arise if for each RKHS we had many possible kernel functions. These difficulties do not occur, as by the Moore-Aronszajn theorem (Theorem 4), the relationship between each kernel and its RKHS is unique. This means that by defining the kernel, we define the set of possible functions from which we can choose our solution.

Theorem 4 (Moore-Aronszajn) *(Theorem 1.1.1 [Wahba, 1990]) To every RKHS there corresponds a unique positive semidefinite function (called the reproducing kernel) and conversely, given a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, we can construct a unique RKHS of real valued functions on $\mathcal{X}$ with k as its reproducing kernel.*

In summary, learning in a RKHS ensures that we have a continuous evaluation functional and we can define the class of functions we are looking at by the kernel function. The kernel also has the added benefit of having an intuitive interpretation of being a mapping into some feature space where a linear algorithm can be applied. For further details of reproducing kernel Hilbert spaces, the reader is referred to Aronszajn [1950], Saitoh [1988] and Wahba [1990].

1.1.2 Regularization

In addition to being able to evaluate our function, we would like a solution from applying our learning algorithm to noisy data to be "near" the optimal solution without noise. There are two parts to measuring the success of our algorithm given some training data. First, we would like to match the training data as closely as possible. However, keeping in mind that there may be noise in the training data, we would like to prevent overfitting by restricting the complexity of the functions we are studying. Regularization aims to balance these two conflicting requirements.

We measure the "nearness" of a function f at an example x_{test} given a label y_{test} by a loss function

$$\ell(x_{\text{test}}, y_{\text{test}}, f(x_{\text{test}})).$$

The loss function $\ell : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$ is defined for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ such that $\ell(x, y, y) = 0$. The notion of risk measures the success of our algorithm and is a combination of these loss functions. We assume that the examples are independently and identically drawn from a distribution $\Pr(x, y)$, and hence we would like to minimize the expected risk,

$$R(f) = \mathbb{E}_{\mathcal{X}\times\mathcal{Y}}(\ell(x, y, f(x))) = \int_{\mathcal{X}\times\mathcal{Y}} \ell(x, y, f(x))d\Pr(x, y).$$

Since the distribution is unknown there are two approaches we can take [Vapnik, 1982, Chapter 2]. The most common approach is the idea of empirical risk minimization with an associated regularization term. The second approach connects the minimization of expected risk with the use of an iterative procedure. This second approach leads to the idea of semi-convergence and regularization by early stopping.

There is a large body of work on empirical risk minimization and associated generalization error bounds. No attempt shall be made to survey this work, and the interested reader is directed to Vapnik [1995, 1998] for the classical approach, Devroye et al. [1996] for a statistical viewpoint and Mendelson [2003] for a tutorial on recent approaches to generalization error bounds.

The advantage of optimization in a RKHS is that under certain conditions the optimal solutions can be found as the linear combination of a finite number of basis functions, regardless of the dimensionality of the space $\mathcal{H}$ the optimization is carried out in. The theorem below formalizes this notion (see [Kimeldorf and Wahba, 1971] and [Schölkopf and Smola, 2002, Theorem 4.2]).

Theorem 5 (Representer Theorem) *Let $\Omega : [0, \infty) \to \mathbb{R}$ be a strictly monotonic increasing function, $\mathcal{X}$ a set, and $\ell : (\mathcal{X} \times \mathbb{R}^2)^m \to \mathbb{R} \cup \{\infty\}$ an arbitrary loss function. Then each minimizer $f \in \mathcal{H}$ of the general regularized risk*

$$\ell\left((x_1, y_1, f(x_1)), \ldots, (x_m, y_m, f(x_m))\right) + \Omega\left(\|f\|_{\mathcal{H}}\right) \tag{1.2}$$

admits a representation of the form

$$f(x) = \sum_{i=1}^{m} \alpha_i k(x_i, x). \tag{1.3}$$

where k is the reproducing kernel of $\mathcal{H}$, and $\alpha_i \in \mathbb{R}$ for all $i = 1, \ldots, m$.

The representer theorem tells us that when performing empirical risk minimization with a regularization term $\Omega\left(\|f\|_{\mathcal{H}}\right)$, the optimal function can be expressed in terms of kernel functions on our training data. Note that the kernel controls the regularization properties of our optimization problem since the regularization term is a function of the norm in $\mathcal{H}$. More general versions of the representer theorem exists, for example when learning in Banach spaces [Micchelli and Pontil, 2004].

1.1.3 Optimization

An important ingredient of a successful machine learning algorithm is a fast and efficient method for computation. As was mentioned earlier, kernels implicitly define a feature mapping to a RKHS. In this RKHS we search for linear hypotheses. One very successful method for finding a good linear hypothesis is the Support Vector Machine (SVM) [Bennett and Mangasarian, 1992, Cortes and Vapnik, 1995]. SVMs are an example of a class of problems which have a global minimum, called convex optimization problems [Rockafellar, 1996, Boyd and Vandenberghe, 2004]. Algorithms for solving convex problems are efficient, which means that large scale problems can be solved.

In addition to being convex and efficient, SVMs only access the training data via their inner products. Hence SVMs are kernel methods and one can replace the inner product by the kernel function (this is commonly referred to as the "kernel trick") and generalise the SVM to nonlinear hypotheses. The success of SVMs have led to proposals for different versions for binary classification [Cortes and Vapnik, 1995, Schölkopf et al., 2000], regression [Smola and Schölkopf, 1998], multiclass classification [Rätsch et al., 2002] and novelty detection [Schölkopf et al., 2001].

Kernel methods are not restricted to SVMs, since the "kernel trick" can be applied to any algorithm that accesses data via inner products. A literature survey of the specific approaches used in this thesis are presented in the relevant sections.

1.2 Contributions of this Thesis

This thesis presents two extensions to the framework for machine learning with kernels. The first is an application of kernels to the problem of learning the kernel itself. This gives rise to the idea of a hyperkernel. The second is a generalization of machine learning with kernels that are not necessarily positive semidefinite. For each extension, we present:

- Motivations for the extension in terms of existing problems in machine learning (Section 2.3 and Section 4.1).
- The functional framework of the kernel involved, and how it relates to the classical reproducing kernel Hilbert space (Section 2.4 and Section 4.2).
- Examples of kernels in the extended class (Section 2.5 and Section 4.2.4).
- An investigation into how to regularize the solution (Section 2.4.1 and Chapter 5).
- Derivations of the optimization problems associated with common machine learning applications in the new framework (Section 3.1 and Chapter 6).
- Experimental evidence that the algorithms solve the learning problems (Section 3.2 and Section 6.2).

Many of the techniques used are well known, some of which may not be well known within the machine learning community but are popular in other fields of science and mathematics. The following sections clarify the novel parts of this thesis and the methods borrowed from other fields.

1.2.1 Learning the Kernel

The first contribution is a framework to learn the best kernel from the training data for a particular estimation task (Chapter 2). The proposed criteria for success (Section 2.3) generalize several traditional ways of choosing kernels. The functional framework (Section 2.4) defines a kernel on the space of kernels itself, hence giving a natural extension to the idea of regularization.

The semidefinite programs corresponding to several algorithms for binary classification, regression and novelty detection were derived, and the actual numerical solution was done using known techniques in convex optimization (Chapter 3).

1.2.2 Learning with Indefinite Kernels

By treating the problem of learning the kernel as an application of RKHS theory, the representer theorem (Theorem 10) implies that the solution is a linear combination of kernels. Note that a linear combination of positive semidefinite kernels is not necessarily positive semidefinite. This means that unless we have further constraints, the notion of a non-positive kernel arises naturally. These non-positive kernels are called indefinite kernels, and they have appeared in several other applications of mathematics and engineering.

The second contribution of this thesis is a framework for machine learning with indefinite kernels. Chapter 4 shows that the idea of an indefinite kernel results in a reproducing kernel Kreĭn space (RKKS), and collects several results from functional analysis to parallel the results for RKHS in the positive semidefinite case. Due to the geometry of Kreĭn spaces, the corresponding notion to minimization in Hilbert space is stabilization. The representer theorem and the intuition behind the resulting optimization problem are presented in Section 4.3, and generalization bounds are computed in Section 4.4. To illustrate the subtle differences between learning in Kreĭn spaces and Hilbert spaces, an investigation of the spectrum of the evaluation operator is shown in Section 4.5.

1.2.3 Regularization by Early Stopping

We argue that the traditional regularized risk minimization framework may not succeed for indefinite kernels, and propose using regularization by early stopping of subspace iterations in Chapter 5. In fact, this regularization paradigm can be applied to positive kernels as well, since they are just a special case. The third contribution of this thesis is a study of regularization using Krylov subspace algorithms. This has been studied in numerical optimization, but is relatively unknown in the machine learning community. Krylov subspace algorithms, which are also known as conjugate gradient type algorithms, are introduced in Section 5.2. Their regularization properties are analysed via the filter functions induced on the spectrum of the kernel in Section 5.3.

Iterative methods such as conjugate gradient type methods applied to ill-posed problems exhibit semi-convergence: the iterates initially converge towards the true solution but as the number of iterations increase, diverge away. The behaviour of a particular algorithm, called Minimal Residual, is analysed in Section 5.4. The results from numerical mathematics are interpreted in terms of machine learning, demonstrating that early stopping indeed performs regularization.

1.3 Summary of Contributions

In summary, the three contributions of this thesis are: a framework for learning the kernel, an extension of kernel methods to indefinite kernels, and an investigation of regularization by early stopping. These contributions can also be viewed independently of each other, as each advance can be applied to areas other than those investigated here.

Chapter 2

Learning the Kernel

This chapter addresses the problem of choosing a kernel suitable for estimation with a Support Vector Machine. This goal is achieved by defining a reproducing kernel Hilbert space on the space of kernels itself. Such a formulation leads to a statistical estimation problem similar to the problem of minimizing a regularized risk functional. The solution of the resulting optimization problem is discussed in Chapter 3.

After a brief introduction and motivation to the need for learning the kernel (Section 2.1, we review methods for model selection (Section 2.2). We show that for kernel-based learning methods there exists a functional, the *quality functional*, which plays a similar role to the empirical risk functional (Section 2.3). We introduce a kernel on the space of kernels itself, a *hyperkernel* (Section 2.4), and its regularization on the associated Hyper reproducing kernel Hilbert space (Hyper-RKHS). This leads to a systematic way of parameterizing kernel classes while managing overfitting. We give several examples of hyperkernels and recipes to construct others (Section 2.5).

2.1 Introduction

Kernel methods have been highly successful in solving various problems in machine learning. The algorithms work by implicitly mapping the inputs into a feature space, and finding a suitable hypothesis in this new space. In the case of the Support Vector Machine (SVM), this solution is the hyperplane which maximizes the margin in the feature space. The feature mapping in question is defined by a kernel function, which allows us to compute dot products in feature space using only objects in the input space. For an introduction to SVMs and kernel methods, the reader is referred to several tutorials (such as Burges [1998]) and books (such as Schölkopf and Smola [2002]).

Choosing a suitable kernel function, and therefore a feature mapping, is imperative to the success of this inference process. To date there are few systematic techniques to assist in this choice.

As motivation for the need for methods to learn the kernel function, consider Figure 2.1, which shows the separating hyperplane and the margin for the same dataset. Figure 2.1(a) shows the classification function for a support vector machine using a Gaussian radial basis function (RBF) kernel. The data has been generated using two Gaussian distributions with standard deviation 1 in one dimension and 1000 in the other. This difference in scale creates problems for the Gaussian RBF kernel, since it is unable to find a kernel width suitable for both directions. Hence, the classification function is dominated by the dimension with large variance. Increasing the value of the regularization parameter C, and hence decreasing the smoothness of the function, results in a hyperplane which is more complex, and equally unsatisfactory (Figure 2.1(b)). The traditional way to handle such data is to normalise each dimension independently.

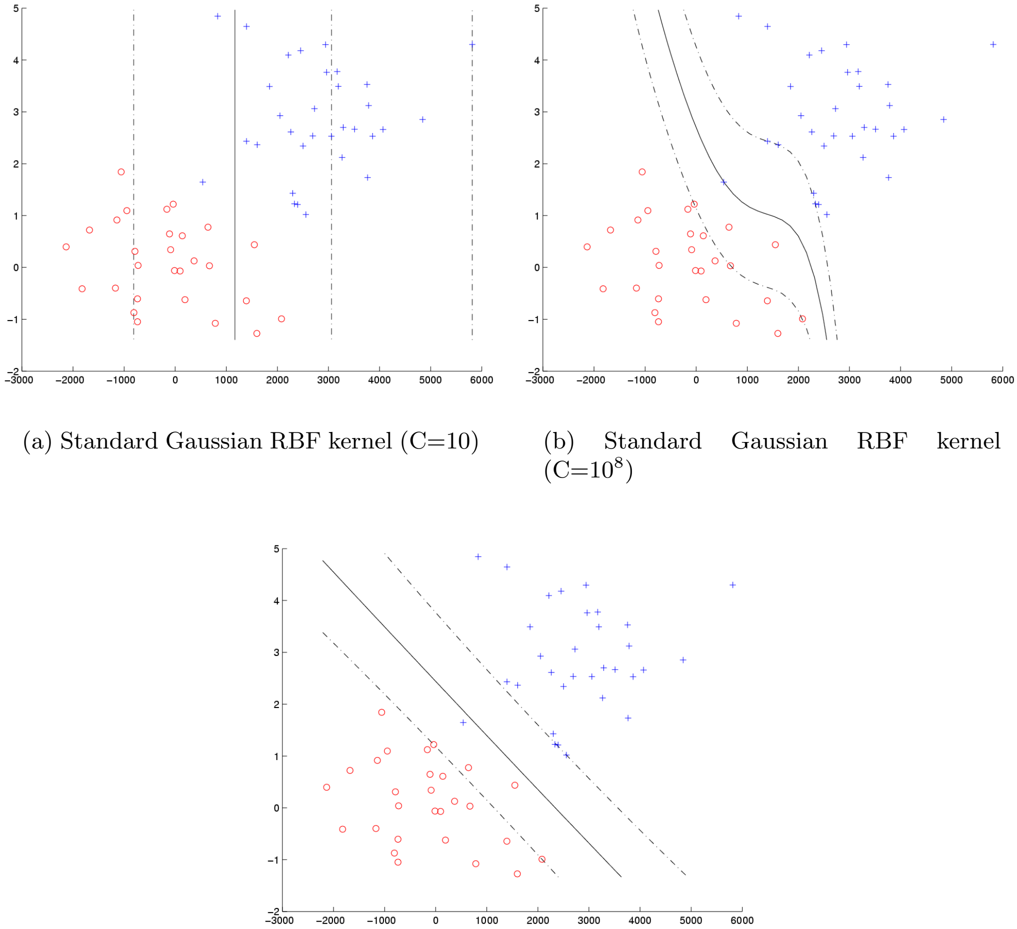
Instead of normalising the input data, we make the kernel adaptive to allow independent scales for each dimension. This allows the kernel to handle unnormalised data. However, the resulting kernel would be difficult to hand-tune as there may be numerous free variables. In this case, we have a free parameter for each dimension of the input. We 'learn' this kernel by defining a quantity analogous to the risk functional, called the quality functional, which measures the 'badness' of the kernel function. The classification function for the above mentioned data is shown in Figure 2.1(c). Observe that it captures the scale of each dimension independently. In general, the solution does not consist of only a single kernel but a linear combination of them.

2.2 Review of Methods for Model Selection

A user of machine learning algorithms is faced with two levels of model selection. Firstly, the user has to decide on the type of algorithm, such as neural networks, Bayesian learning, maximum likelihood estimation or SVM. Then, within that particular class of algorithms, the user has to choose a particular setting to solve a particular problem. Learning the kernel falls into the latter case.

2.2.1 Model Selection for Machine Learning

Model selection inherently requires certain assumptions about the problem. This is formalized in the No Free Lunch theorem [Wolpert, 2001], which asserts that there is no algorithm which performs better than all other algorithms on all datasets. An analogous theorem for feature representation called the Ugly Duckling Theorem [Watanabe, 1985] asserts that there is no best representation for data, and even the notion of similarity between objects depends on assumptions. We shall explicitly state the assumptions made for previous work on learning the kernel.



(a) Standard Gaussian RBF kernel (C=10)

(b) Standard Gaussian RBF kernel (C=10^8)

(c) RBF-Hyperkernel with adaptive widths

Figure 2.1: Plot of synthetic data, showing the separating hyperplane and the margins given for a uniformly chosen length scale (top) and an automatic width selection (bottom). For data with highly non-isotropic variance, choosing one scale for all dimensions leads to unsatisfactory results.

2.2.2 Learning the Kernel

We analyze some recent approaches to learning the kernel by looking at the objective function that is being optimized and the class of kernels being considered. We will see later (Section 2.3) that this objective function is related to our definition of a quality functional. We denote by K the kernel matrix given by $K_{ij} := k(x_i, x_j)$ where $x_i, x_j \in \mathcal{X}$. We also use $\mathrm{tr} K$ to mean the trace of the matrix and $\det K$ to mean the determinant.

Cross validation has been used to select the parameters of the kernels and SVMs [Duan et al., 2003, Meyer et al., 2003], with varying degrees of success. The objective function is the regularized risk functional, and the class of kernels is a finite set defined by a grid of the possible parameter settings. Duan et al. [2003] and Chapelle et al. [2002] tests various approximations which bound the leave one out error, or some measure of the capacity of the SVM. The notion of alignment [Cristianini et al., 2003] can be seen as an instance of the empirical quality functional. Hence the objective function is $\mathrm{tr}(Kyy^\top)$ where y are the training labels, and K is from the class of kernels spanned by the eigenvectors of the kernel matrix of the combined training and test data. The SDP approach [Lanckriet et al., 2002, 2004] uses a more general class of kernels, namely a linear combination of positive semidefinite matrices. They minimize the margin of the resulting SVM using a SDP for kernel matrices with constant trace. Similar to this, Bousquet and Herrmann [2002] further restricts the class of kernels to the convex hull of the kernel matrices normalized by their trace. This restriction, along with minimization of the complexity class of the kernel, allows them to perform gradient descent to find the optimum kernel. Using the idea of boosting, Crammer et al. [2002] optimize $\sum_t \beta_t K_t$, where β_t are the weights used in the boosting algorithm. The class of base kernels is obtained from the normalized solution of the generalized eigenvector problem. In principle, one can learn the kernel using Bayesian methods by defining a suitable prior, and learning the hyperparameters by optimizing the marginal likelihood [Williams and Rasmussen, 1996, Williams and Barber, 1998]. As an example of this, when other information is available, an auxiliary matrix can be used with the EM algorithm for learning the kernel [Tsuda et al., 2003]. Table 2.1 summarises these approaches. The notation $K \succeq 0$ means that K is positive semidefinite, that is for all $a \in \mathbb{R}^n$, $a^\top K a \geqslant 0$.

2.3 Kernel Quality Functionals

To optimize over a class of kernels, we need a performance measure which tells us whether we are successful or not. We introduce a new class of functionals Q on data which we will call *quality functionals*. Note that by quality we actually mean *badness* or

Approach	Objective	Kernel class ($\mathcal{K}$)
Cross Validation	CV Risk	Finite set of kernels
Alignment	$y^\top K y$	$\{\sum_{i=1}^m \beta_i v_i v_i^\top$ where v_i are eigenvectors of $K\}$
SDP	margin	$\{\sum_{i=1}^m \beta_i K_i$ s.t. $K_i \succeq 0, \mathrm{tr} K_i = c\}$
Complexity Bound	margin	$\{\sum_{i=1}^m \beta_i K_i$ s.t. $K_i \succeq 0, \mathrm{tr} K_i = c, \beta_i \geqslant 0\}$
Boosting	Exp/LogLoss	Base kernels from eigenvector problem
Bayesian	neg. log-post.	dependent on prior
EM Algorithm	KL Divergence	linear combination of auxiliary matrix

Table 2.1: Summary of recent approaches to kernel learning.

lack of quality, as we would like to minimize this quantity. Their purpose is to indicate, given a kernel k and the training data, how suitable the kernel is for explaining the training data or, in other words, the *quality* of the kernel for the estimation problem at hand. Such quality functionals may be the kernel target alignment, the negative log posterior, the minimum of the regularized risk functional, or any luckiness function for kernel methods. We will discuss those functionals after a formal definition of the quality functional itself.

2.3.1 Empirical and Expected Quality

Definition 6 (Empirical Quality Functional) *Let $\mathcal{H}_+$ be the class of positive semi-definite functions. Given a kernel k, and data X, Y, we define $Q_{\mathrm{emp}}(k, X, Y) : \mathcal{H}_+ \times \mathcal{X}^m \times \mathcal{Y}^m \to \mathbb{R}$ to be an* empirical quality functional *if it depends on k only via $k(x_i, x_j)$ where $x_i, x_j \in \mathcal{X}$.*

By this definition, Q_{emp} is a function which tells us how well matched k is to a specific dataset X, Y. Typically such a quantity is used to adapt k in such a manner that Q_{emp} is optimal (e.g., optimal alignment, greatest luckiness, smallest negative log-posterior), based on this one *single* dataset X, Y. Provided a sufficiently rich class of kernels $\mathcal{F}$ it is in general possible to find a kernel k^* that attains the minimum of any such Q_{emp} regardless of the data. However, it is very unlikely that $Q_{\mathrm{emp}}(k^*, X, Y)$ would be similarly small for other X, Y, for such a k^*. To measure the overall quality of k we therefore introduce the following definition:

Definition 7 (Expected Quality Functional) *Denote by $Q_{\mathrm{emp}}(k, X, Y)$ an empirical quality functional, then*

$$Q(k) := \mathbb{E}_{X,Y} \left[Q_{\mathrm{emp}}(k, X, Y) \right] \tag{2.1}$$

is defined to be the expected quality functional. *Here the expectation is taken over*

X, Y, where all x_i, y_i are drawn from $\Pr(x, y)$.

Observe the similarity between the empirical quality functional, $Q_{\text{emp}}(k, X, Y)$, and the empirical risk of an estimator, $R_{\text{emp}}(f, X, Y) = \frac{1}{m}\sum_{i=1}^{m} \ell(x_i, y_i, f(x_i))$ (where ℓ is a suitable loss function); in both cases we compute the value of a functional which depends on some sample X, Y drawn from $\Pr(x, y)$ and a function. We have

$$Q(k) = \mathbb{E}_{X,Y}\left[Q_{\text{emp}}(k, X, Y)\right] \text{ and } R(f) = \mathbb{E}_{X,Y}\left[R_{\text{emp}}(f, X, Y)\right]. \tag{2.2}$$

Here $R(f)$ denotes the expected risk. However, while in the case of the empirical risk, we can interpret R_{emp} as the the empirical estimate of the expected loss $R(f) = \mathbb{E}_{x,y}[\ell(x, y, f(x))]$, no such analogy is available for quality functionals, due to the general form of Q_{emp}.

Finding a general-purpose bound of the expected error in terms of $Q(k)$ is difficult, since the definition of Q depends heavily on the algorithm under consideration. Nonetheless, it provides a general framework within which such bounds can be derived.

To obtain a generalization error bound, one would require that Q_{emp} is concentrated around its expected value. Furthermore, one would require the deviation of the empirical risk to be bounded above by Q_{emp} and possibly other terms. In other words, we make the following two assumptions: we have given a concentration inequality on quality functionals, such as

$$\Pr\left\{|Q_{\text{emp}}(k, X, Y) - Q(k)| \geqslant \varepsilon_Q\right\} < \delta_Q,$$

and we have a bound on the deviation of the empirical risk in terms of the quality functional

$$\Pr\left\{|R_{\text{emp}}(f, X, Y) - R(f)| \geqslant \varepsilon_R\right\} < \delta(Q_{\text{emp}}).$$

Then we can chain both inequalities together to obtain the following bound

$$\Pr\left\{|R_{\text{emp}}(f, X, Y) - R(f)| \geqslant \varepsilon_R\right\} < \delta_Q + \delta(Q + \varepsilon_Q).$$

This means that the bound now becomes independent of the particular value of the quality functional obtained *on* the data, rather than the expected value of the quality functional. Bounds of this type have been derived for Kernel Target Alignment [Cristianini et al., 2003, Theorem 9] and the Algorithmic Luckiness framework [Herbrich and Williamson, 2002, Theorem 17].

2.3.2 Examples of Q_{emp}

Before we continue with the derivations of a regularized quality functional and introduce a corresponding reproducing kernel Hilbert space, we give some examples of quality functionals and present their exact minimizers, whenever possible. This demonstrates that given a rich enough feature space, we can arbitrarily minimize the empirical quality functional Q_{emp}. The difference here from traditional kernel methods is the fact that we allow the kernel to change. This extra degree of freedom allows us to overfit the training data. In many of the examples below, we show that given a feature mapping which can model the labels of the training data precisely, overfitting occurs. That is, if we use the training labels as the kernel matrix, we arbitrarily minimize the quality functional. The reader who is convinced that one can arbitrarily minimize Q_{emp} by optimizing over a suitably large class of kernels, may skip the following examples.

Example 1 (Regularized Risk Functional) *These are commonly used in SVMs and related kernel methods (see for example Wahba [1990], Vapnik [1995] and Schölkopf and Smola [2002]). They take on the general form*

$$R_{\text{reg}}(f, X_{\text{train}}, Y_{\text{train}}) := \frac{1}{m}\sum_{i=1}^{m} \ell(x_i, y_i, f(x_i)) + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2 \tag{2.3}$$

where $\|f\|_{\mathcal{H}}^2$ is the RKHS norm of f. By virtue of the representer theorem (see Section 2.4) we know that the minimizer of (2.3) can be written as a kernel expansion. This leads to the following definition of a quality functional, for a particular cost functional ℓ:

$$Q_{\text{emp}}^{\text{regrisk}}(k, X_{\text{train}}, Y_{\text{train}}) := \min_{\alpha \in \mathbb{R}^m} \left[\frac{1}{m}\sum_{i=1}^{m} \ell(x_i, y_i, [K\alpha]_i) + \frac{\lambda}{2}\alpha^\top K\alpha\right]. \tag{2.4}$$

We now construct examples that minimize Equation (2.4).

- *First, note that for $K = \beta yy^\top$ and $\alpha = \frac{1}{\beta\|y\|^2}y$ we have $K\alpha = y$ and $\alpha^\top K\alpha = \beta^{-1}$. This leads to $Q_{\text{emp}}^{\text{regrisk}}(k, X_{\text{train}}, Y_{\text{train}}) = \frac{\lambda}{2\beta}$. For sufficiently large β we can make $Q_{\text{emp}}^{\text{regrisk}}(k, X_{\text{train}}, Y_{\text{train}})$ arbitrarily close to 0.*

- *Even if we disallow setting K arbitrarily close to zero by setting $\operatorname{tr}K = 1$, finding the minimum of (2.4) can be achieved as follows: let $K = \frac{1}{\|z\|^2}zz^\top$, where $z \in \mathbb{R}^m$, and $\alpha = z$. Then $K\alpha = z$ and we obtain*

$$\frac{1}{m}\sum_{i=1}^{m} l(x_i, y_i, [K\alpha]_i) + \frac{\lambda}{2}\alpha^\top K\alpha = \sum_{i=1}^{m} l(x_i, y_i, z_i) + \frac{\lambda}{2}\|z\|_2^2. \tag{2.5}$$

Choosing each $z_i = \operatorname{argmin}_\zeta\, l(x_i, y_i, \zeta(x_i)) + \frac{\lambda}{2}\zeta^2$, where ζ are the possible hypoth-

esis functions obtained from the training data, yields the minimum with respect to z. Since $y_i = z_i$ for all i, (2.5) tends to zero and the regularized risk is lower bounded by zero, we can still arbitrarily minimize $Q_{\text{emp}}^{\text{regrisk}}$.

Example 2 (Negative Log-Posterior) *This functional is similar to R_{reg}, as it includes a regularization term (in this case the negative log prior), a loss term (the negative log-likelihood), and additionally, the log-determinant of K [Schölkopf and Smola, 2002, Chapter 16]. The latter measures the size of the space spanned by K. This leads to the following quality functional:*

$$Q_{\text{emp}}^{\text{logpos}}(k, X_{\text{train}}, Y_{\text{train}}) := \min_{f \in \mathbb{R}^m} \left[-\log p(y_i | x_i, f_i) + \frac{1}{2} f^\top K^{-1} f + \frac{1}{2} \log \det K \right] \quad (2.6)$$

This quality functional also has a non meaningful minimizer. Note that any K which does not have full rank will send (2.6) to $-\infty$, hence Q_{emp} is minimized trivially. If we fix the determinant of K to be some constant to ensure that K is full rank, we can set

$$K = \beta \|y\|^{-2} yy^\top + \beta^{-\frac{1}{m-1}} (\mathbf{1} - \|y\|^{-2} yy^\top) \quad (2.7)$$

which leads to $|K| = 1$. Under the assumption that the minimum of $-\log p(y_i, x_i, f_i)$ with respect to f_i is attained at $f_i = y_i$, we can see that $\beta \longrightarrow \infty$ leads to the overall minimum of $Q_{\text{emp}}^{\text{logpos}}(k, X_{\text{train}}, Y_{\text{train}})$.

Example 3 (Cross Validation) *Cross validation is a widely used method for estimating the generalization error of a particular learning algorithm. Specifically, the leave-one-out cross validation is an almost unbiased estimate of the generalization error [Luntz and Brailovsky, 1969]. The quality functional for classification using kernel methods is given by:*

$$Q_{\text{emp}}^{\text{loo}}(k, X_{\text{train}}, Y_{\text{train}}) = \min_{\alpha \in \mathbb{R}^m} \left[\frac{1}{m} \sum_{i=1}^{m} -y_i \text{sign}([K\alpha]_i) \right], \quad (2.8)$$

which was optimized in Duan et al. [2003] and Meyer et al. [2003].

Choosing $K = yy^\top$ and $\alpha^i = \frac{1}{\|y^i\|^2} y^i$, where α^i and y^i are the vectors with the ith element set to zero, we have $K\alpha = y$. Hence we can achieve perfect prediction. For a validation set of larger size, i.e. k-fold cross validation, the same result can be achieved by defining a corresponding α.

Example 4 (Kernel Target Alignment) *This quality functional was introduced by Cristianini et al. [2001] to assess the alignment of a kernel with training labels. It is* *defined by*

$$Q_{\text{emp}}^{\text{alignment}}(k, X_{\text{train}}, Y_{\text{train}}) := 1 - \frac{y^\top K y}{\|y\|_2^2 \|K\|_2}. \tag{2.9}$$

Here $\|y\|_2$ denotes the ℓ_2 norm of the vector of observations and $\|K\|_2$ is the Frobenius norm, i.e., $\|K\|_2^2 := \operatorname{tr} KK^\top = \sum_{i,j} K_{ij}^2$. This quality functional was optimized in Lanckriet et al. [2004].

We show that this quality functional is not immune to overfitting. By decomposing K into its eigensystem we see that (2.9) is minimized, if $K = yy^\top$, in which case

$$Q_{\text{emp}}^{\text{alignment}}(k^*, X_{\text{train}}, Y_{\text{train}}) = 1 - \frac{y^\top y y^\top y}{\|y\|_2^2 \|yy^\top\|_2} = 1 - \frac{\|y\|_2^4}{\|y\|_2^2 \|y\|_2^2} = 0. \tag{2.10}$$

It is clear that we cannot expect that $Q_{\text{emp}}^{\text{alignment}}(k^*, X, Y) = 0$ for data other than that chosen to determine k^*, in other words, a restriction of the class of kernels is required.

Example 5 (Luckiness for Classification with Kernels) *Recently the concept of algorithmic luckiness [Herbrich and Williamson, 2002] was introduced to assess the quality of an estimate in a sample and algorithm dependent fashion. We define the quality functional for a kernel method to be:*

$$Q_{\text{emp}}^{\text{luckiness}}(k, X_{\text{train}}, Y_{\text{train}}) := \min_{j \in \mathbb{N}} \left\{ j \geqslant \left(\frac{\varepsilon_j(X_{\text{train}}) \|\alpha\|_1}{\Gamma_{(X_{\text{train}}, Y_{\text{train}})}(w_\alpha)} \right)^2 \right\}$$

where $\varepsilon_j(X_{\text{train}})$ is the smallest ε such that $\{\phi(x_1), \ldots, \phi(x_n)\}$ can be covered by at most j balls of radius ε, α is the vector (dual coefficients) of the maximum margin solution, w_α, is the corresponding weight vector, ϕ is the feature mapping corresponding to k, and $\Gamma_{(X_{\text{train}}, Y_{\text{train}})}(w_\alpha)$ is the normalized margin $\min_{(x,y) \in (X_{\text{train}}, Y_{\text{train}})} \frac{y_i \langle \phi(x_i), w_\alpha \rangle}{\|\phi(x_i)\| \|w_\alpha\|}$.

For $K = yy^\top$, we can cover the feature space by balls of radius 1, that is $\varepsilon_j(X_{\text{train}}) \leqslant 1$ for all $j \geqslant 2$. Since the algorithmic luckiness framework depends on the choice of a particular algorithm, we have to choose a rule for generating α. We consider any choice for which $y_i \alpha_i \geq 0$ and $\|\alpha\|_1 = 1$, as is satisfied for SVM, linear programming estimators, and many boosting algorithms. For this choice, the empirical error vanishes with margin 1 and by construction $\|\alpha\|_1 = 1$. Hence, $Q_{\text{emp}}^{\text{luckiness}}(k, X_{\text{train}}, Y_{\text{train}}) = 1$, which is the global minimum.

Example 6 (Radius-Margin Bound) *For SVMs without thresholding and with no training errors, Vapnik [1998] proposed the following upper bound on the generalization error of the classifier in terms of the radius and margin of the SVM [Bartlett and Shawe-Taylor, 1999].*

$$T = \frac{1}{m}\frac{R^2}{\gamma^2} \tag{2.11}$$

where R and γ are the radius and the margin of the training data. We can define a quality functional:

$$Q^{\text{radius}}_{\text{emp}}(k, X_{\text{train}}, Y_{\text{train}}) = \frac{1}{m}R^2\alpha^\top K\alpha, \tag{2.12}$$

which was optimized in Chapelle et al. [2002].

We find the conditions were this quality functional is minimized in a non meaningful manner. Choosing $K = \beta yy^\top$ and $\alpha = \frac{1}{\beta\|y\|^2}y$, we obtain a bound on the radius $R^2 \leqslant \beta(\max_i y_i^2)$, and an expression for the margin, $\alpha^\top K\alpha = \beta^{-1}$. Therefore $Q^{\text{radius}}_{\text{emp}}(k, X_{\text{train}}, Y_{\text{train}}) \leqslant \frac{\beta^2}{m}$, which can be made arbitrarily close to zero by letting $\beta \longrightarrow 0$.

The above examples illustrate how many existing methods for assessing the quality of a kernel fit within the quality functional framework. We also saw that given a rich enough class of kernels $\mathcal{F}$, optimization of Q_{emp} over $\mathcal{F}$ would result in a kernel that would be useless for prediction purposes, in the sense that they can be made to look arbitrarily good in terms of Q_{emp} but with the result that the generalization performance will be poor. This is yet another example of the danger of optimizing too much and overfitting – there is (still) no free lunch.

2.4 Hyper Reproducing Kernel Hilbert Spaces

We now propose a method to optimize quality functionals over classes of kernels by introducing a reproducing kernel Hilbert space *on the kernel k itself*, so to say, a Hyper-RKHS. This is really just a RKHS with additional conditions on the elements. In fact, we can have a recursive definition of an RKHS of an RKHS ad infinitum.

2.4.1 Regularized Quality Functional

We define an RKHS on kernels $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, simply by introducing the compounded index set, $\underline{\mathcal{X}} := \mathcal{X} \times \mathcal{X}$ and by treating k as functions $k : \underline{\mathcal{X}} \to \mathbb{R}$:

Definition 8 (Hyper Reproducing Kernel Hilbert Space) *Let $\mathcal{X}$ be a nonempty set and denote by $\underline{\mathcal{X}} := \mathcal{X} \times \mathcal{X}$ the compounded index set. The Hilbert space $\underline{\mathcal{H}}$ of functions $k : \underline{\mathcal{X}} \to \mathbb{R}$, endowed with a dot product $\langle \cdot, \cdot \rangle$ (and the norm $\|k\| = \sqrt{\langle k, k \rangle}$) is called a Hyper Reproducing Kernel Hilbert Space if there exists a hyperkernel $\underline{k} : \underline{\mathcal{X}} \times \underline{\mathcal{X}} \to \mathbb{R}$ with the following properties:*

1. *$\underline{k}$ has the reproducing property*

$$\langle k, \underline{k}(\underline{x}, \cdot)\rangle = k(\underline{x}) \text{ for all } k \in \underline{\mathcal{H}}; \tag{2.13}$$

in particular, $\langle \underline{k}(\underline{x}, \cdot), \underline{k}(\underline{x}', \cdot)\rangle = \underline{k}(\underline{x}, \underline{x}')$.

2. *$\underline{k}$ spans $\underline{\mathcal{H}}$, i.e. $\underline{\mathcal{H}} = \overline{\mathrm{span}\{\underline{k}(\underline{x}, \cdot)|\underline{x} \in \underline{\mathcal{X}}\}}$.*

3. *$\underline{k}(x, y, s, t) = \underline{k}(y, x, s, t)$ for all $x, y, s, t \in \mathcal{X}$.*

This is a RKHS with the additional requirement of symmetry in its first two arguments. We define the corresponding notations for elements, kernels, and RKHS by underlining it. What distinguishes $\underline{\mathcal{H}}$ from a normal RKHS is the particular form of its index set ($\underline{\mathcal{X}} = \mathcal{X}^2$) and the additional condition on $\underline{k}$ to be symmetric in its second two arguments. It is implicit in the definition that the hyperkernel is symmetric between the pairs of arguments (since it is a kernel), that is $\underline{k}(x, y, s, t) = \underline{k}(s, t, x, y)$ for all (x, y) and $(s, t) \in \underline{\mathcal{H}}$, and hence $\underline{k}$ is a kernel in its first two arguments as well.

This approach of defining a RKHS on the space of symmetric functions of two variables leads us to a natural regularization method. By analogy with the definition of the regularized risk functional (2.3), we proceed to define the regularized quality functional

Definition 9 (Regularized Quality Functional) *Let X, Y be the combined training and test set of examples and labels respectively and Q_{emp} an empirical quality functional defined on the whole of $\underline{\mathcal{H}}$, satisfying the conditions of Theorem 5. For a positive semidefinite kernel matrix K on X, the* regularized quality functional *is defined as*

$$Q_{\mathrm{reg}}(k, X, Y) := Q_{\mathrm{emp}}(k, X, Y) + \frac{\lambda_Q}{2}\|k\|^2_{\underline{\mathcal{H}}}, \tag{2.14}$$

where $\lambda_Q \geqslant 0$ is a regularization constant and $\|k\|^2_{\underline{\mathcal{H}}}$ denotes the RKHS norm in $\underline{\mathcal{H}}$.

Note that although we have possibly non positive kernels in $\underline{\mathcal{H}}$, we define the regularized quality functional only on positive semidefinite kernel matrices. This is a slightly weaker condition than requiring a positive semidefinite kernel k, since we only require positivity on the data. Since Q_{emp} depends on k only via the data, this is sufficient for the above definition. Minimization of Q_{reg} is less prone to overfitting than minimizing Q_{emp}, since the regularization term $\frac{\lambda_Q}{2}\|k\|^2_{\underline{\mathcal{H}}}$ effectively controls the complexity of the class of kernels under consideration. The complexity of this class can be derived from the results of Bousquet and Herrmann [2002]. Regularizers other than $\frac{\lambda_Q}{2}\|k\|^2_{\underline{\mathcal{H}}}$ are possible, such as ℓ_p penalties. In this paper, we restrict ourselves to the ℓ_2 norm (2.14). The advantage of (2.14) is that its minimizer satisfies the representer theorem.

Corollary 10 (Representer Theorem for Hyper-RKHS) *Let $\mathcal{X}$ be a set, Q_{emp} an empirical quality functional defined on the whole of $\underline{\mathcal{H}}$, satisfying the conditions of Theorem 5, and X, Y the combined training and test set, then each minimizer $k \in \underline{\mathcal{H}}$ of the regularized quality functional $Q_{\mathrm{reg}}(k, X, Y)$ admits a representation of the form*

$$k(x, x') = \sum_{i,j}^{m} \beta_{ij} \underline{k}((x_i, x_j), (x, x')) \text{ for all } x, x' \in X, \tag{2.15}$$

where $\beta_{ij} \in \mathbb{R}$ for $i, j \in 1, \ldots m$.

Proof All we need to do is rewrite (2.14) so that it satisfies the conditions of Theorem 5. Let $\underline{x}_{ij} := (x_i, x_j)$. Then $Q_{\mathrm{emp}}(k, X, Y)$ has the properties of a loss function, as it only depends on k via its values at $\underline{x}_{ij}$. Note too that the kernel matrix K also only depends on k via its values at $\underline{x}_{ij}$. Furthermore, $\frac{\lambda_Q}{2} \|k\|^2_{\underline{\mathcal{H}}}$ is an RKHS regularizer, so the representer theorem applies and (2.15) follows. ■

Corollary 10 implies that the solution of the regularized quality functional is a linear combination of hyperkernels on the input data. This shows that even though the optimization takes place over an entire Hilbert space of kernels, one can find the optimal solution by choosing among a finite number.

Although Corollary 10 provides the form for the minimizer of the regularized quality functional Q_{reg}, the minimizer given by Equation (2.15) is not necessarily positive semidefinite. This is because β_{ij} can be negative and we get a possibly negative kernel in the summation. Furthermore, we do not have any guarantees that the kernel will be positive semidefinite for examples in the test set, since our restriction in Equation (2.14) is only on the kernel matrix. While there has been some work on learning with indefinite kernels (for example Goldfarb [1985], Haasdonk [2003], Mary [2003]), the majority of applications require positive semidefinite kernels. Hence we focus on learning positive semidefinite kernels. The extension to learning with indefinite kernels is discussed in Chapter 4.

In the following, we impose a positivity restriction on the coefficients of the hyperkernel expansion, that is for all $i, j \in 1, \ldots, m$, we require $\beta_{i,j} \geqslant 0$. The condition of positive expansion coefficients is sufficient but not necessary for the positivity of the kernel given the hyperkernel. In actual fact, we need to impose constraints of the type $K \succeq 0$ or k is a Mercer Kernel. While the latter is almost impossible to enforce directly, the former could be verified directly, hence imposing a constraint only on the values of the kernel matrix $k(x_i, x_j)$ rather than on the kernel function k itself. This means that the conditions of the representer theorem apply with suitable constraints on the coefficients β_{ij}. Given that $\underline{k}$ is positive semidefinite, and we want k to be positive semidefinite, for each of the coefficients $\beta_{11}, \ldots, \beta_{mm}$, there exists a set of α_i

and x_i such that $\sum_{i,j=1}^m \alpha_i \alpha_j k(x_i, x_j) = \beta_{kl}$. Unfortunately, this condition is difficult to enforce in practice.

Another option is to be somewhat more restrictive and require that all expansion coefficients $\beta_{i,j} \geqslant 0$ and all the functions be positive semidefinite kernels. This latter requirement can be formally stated as follows: For any fixed $\underline{x} \in \underline{\mathcal{X}}$ the hyperkernel $\underline{k}$ is a kernel in its second argument; that is for any fixed $\underline{x} \in \underline{\mathcal{X}}$, the function $k(x, x') := \underline{k}(\underline{x}, (x, x'))$, with $x, x' \in \mathcal{X}$, is a positive semidefinite kernel.

Proposition 11 *Given a hyperkernel, $\underline{k}$ with elements such that for any fixed $\underline{x} \in \underline{\mathcal{X}}$, the function $k(x_k, x_l) := \underline{k}(\underline{x}, (x_k, x_l))$, with $x_k, x_l \in \mathcal{X}$, is a positive semidefinite kernel, and $\beta_{ij} \geqslant 0$ for all $i, j = 1, \ldots, m$, then the kernel*

$$k(x_k, x_l) := \sum_{i,j=1}^{m} \beta_{ij} \underline{k}(x_i, x_j, x_k, x_l)$$

is positive semidefinite.

Proof The result is obtained by observing that positive combinations of positive semidefinite kernels are positive semidefinite. ■

While this may prevent us from obtaining the minimizer of the objective function, it yields a much more amenable optimization problem in practice, in particular if the resulting cone spans a large enough space (as happens with increasing m). In the subsequent derivations of optimization problems, we choose this restriction as it provides a more tractable problem in practice. In Section 2.5, we give examples and recipes for constructing hyperkernels. Before that, we relate our framework defined above to Bayesian inference.

2.4.2 A Bayesian Perspective

A generative Bayesian approach to inference encodes all knowledge we might have about the problem setting into a prior distribution. Hence, the choice of the prior distribution determines the behaviour of the inference, as once we have the data, we condition on the prior distribution we have chosen to obtain the posterior, and then marginalize to obtain the label that we are interested in. One popular choice of prior is the normal distribution, resulting in a Gaussian process (GP). All prior knowledge we have about the problem is then encoded in the covariance of the GP. There exists a GP analog to the Support Vector Machine (for example Opper and Winther [2000], Seeger [1999]), which is essentially obtained (ignoring normalizing terms) by exponentiating the regularized risk functional used in SVMs.

In this section, we derive the prior and hyperprior implied by our framework of hyperkernels. This is obtained by exponentiating Q_{reg}, again ignoring normalization terms. Given the regularized quality functional (Equation 2.14), with the Q_{emp} set to the SVM with squared loss, we obtain the following equation.

$$Q_{\text{reg}}(k, X, Y) := \frac{1}{m}\sum_{i=1}^{m}(y_i - f(x_i))^2 + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}} + \frac{\lambda_Q}{2}\|k\|^2_{\underline{\mathcal{H}}}.$$

Exponentiating the negative of the above equation gives,

$$\begin{aligned}&\exp(-Q_{\text{reg}}(k, X, Y)) = \\ &\exp\left(-\frac{1}{m}\sum_{i=1}^{m}(y_i - f(x_i))^2\right)\exp\left(-\frac{\lambda}{2}\|f\|^2_{\mathcal{H}}\right)\exp\left(-\frac{\lambda_Q}{2}\|k\|^2_{\underline{\mathcal{H}}}\right).\end{aligned} \tag{2.16}$$

We compare Equation (2.16) to Gaussian process estimation. The general scheme is known in Bayesian estimation as hyperpriors [Bishop, 1995, Chapter 10], which determine the distribution of the priors (here the GP with covariance k). Figure 2.2 describes the model of an ordinary GP, where f is drawn from a Gaussian distribution with covariance matrix K and y is conditionally independent given f. For hyperprior estimation, we draw the prior K from a distribution instead of setting it.

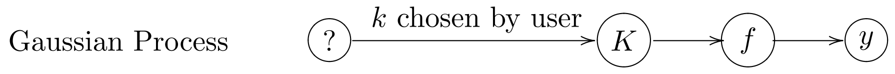


Figure 2.2: Generative model for Gaussian process estimation

To determine the distribution from which we draw the prior, we compute the hyperprior explicitly. For given data $Z = \{X_{\text{train}}, Y_{\text{train}}\}$ and applying Bayes' Rule,

$$\begin{aligned}p(k|Z) &= \frac{p(Z|k)p(k)}{p(Z)} \\ \frac{1}{p(Z|k)} &= \frac{p(k)}{p(k|Z)p(Z)}.\end{aligned}$$

Hence, the posterior is given by

$$\begin{aligned}p(f|Z, k) &= \frac{p(Z|f, k)p(f|k)}{p(Z|k)} \\ &= \frac{p(Z|f, k)p(f|k)p(k)}{p(k|Z)p(Z)}.\end{aligned} \tag{2.17}$$

We have the directed graphical model shown in Figure 2.3 for a Hyperkernel-GP, where we assume that K itself is drawn according to a distribution before performing further steps of dependency calculation. We shall now explicitly compute the terms in the numerator of Equation (2.17).

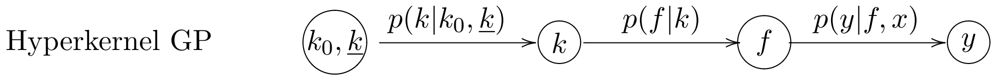


Figure 2.3: Generative model for Gaussian process estimation using hyperpriors on k defined by $\underline{k}$.

In the following derivations, we assume that we are dealing with finite dimensional objects, to simplify the calculations of the normalizing constants in the expressions for the distributions. Given that we have additive Gaussian noise, that is $\epsilon \sim \mathcal{N}(0, \frac{1}{\gamma_\epsilon}\mathbf{I})$, then,

$$p(y|f,x) \propto \exp\left(-\frac{\gamma_\epsilon}{2}(y - f(x))^2\right).$$

Therefore, for the whole dataset (assumed to be i.i.d.),

$$\begin{aligned} p(Y|f,X) &= \prod_{i=1}^{m} p(y_i|f,x_i) \\ &= \left(\frac{2\pi}{\gamma_\epsilon}\right)^{-\frac{m}{2}} \exp\left(-\frac{\gamma_\epsilon}{2}\sum_{i=1}^{m}(y_i - f(x_i))^2\right). \end{aligned}$$

We assume a Gaussian prior on the function f, with covariance function k. The positive semidefinite function, k, defines an inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}_k}$ in the RKHS denoted by $\mathcal{H}_k$. Then,

$$p(f|k) = \left(\frac{2\pi}{\gamma_f}\right)^{-\frac{F}{2}} \exp\left(-\frac{\gamma_f}{2}\langle f, f\rangle_{\mathcal{H}_k}\right)$$

where F is the dimension of f and γ_f is a constant.

We assume a Wishart distribution [Lauritzen, 1996, Appendix C], with p degrees of freedom and covariance k_0, for the prior distribution of the covariance function k, that is $k \sim \mathcal{W}_m(p, k_0)$. This is a hyperprior used in the Gaussian process literature.

$$p(k|k_0) = \frac{|k|^{\frac{p-(m+1)}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}(kk_0)\right)}{\Gamma_m(p)|k|^{\frac{p}{2}}}$$

where $\Gamma_m(p)$ denotes the Gamma distribution, $\Gamma_m(p) = 2^{\frac{pm}{2}}\pi^{\frac{m(m-1)}{4}}\prod_{i=1}^{m}\Gamma\left(\frac{p-i+1}{2}\right)$. For more details of the Wishart distribution, the reader is referred to Lauritzen [1996] and Robert [2001].

Observe that $\mathrm{tr}(kk_0)$ is an inner product between two matrices. We can define a general inner product between two matrices, as the inner product defined in the RKHS denoted by $\underline{\mathcal{H}}$.

$$p(k|k_0, \underline{k}) = \frac{|k|^{\frac{p-(m+1)}{2}} \exp\left(-\frac{1}{2}\langle k, k_0\rangle_{\underline{\mathcal{H}}}\right)}{\Gamma_m(p)|k|^{\frac{p}{2}}}$$

We can interpret the above equation as measuring the similarity between the covariance matrix that we obtain from data and the expected covariance matrix (given by the user). This similarity is measured by a dot product defined by $\underline{k}$. Substituting the expressions for $p(Y|X, f), p(f|k)$ and $p(k|k_0, \underline{k})$ into the posterior (Equation 2.17), we get Equation (2.18) which is of the same form as the exponentiated negative quality (Equation 2.16).

$$\exp\left(-\frac{\gamma_\epsilon}{2}\sum_{i=1}^{m}(y_i - f(x_i))^2\right) \exp\left(-\frac{\gamma_f}{2}\langle f, f\rangle_{\mathcal{H}_k}\right) \exp\left(-\frac{1}{2}\langle k, k_0\rangle_{\underline{\mathcal{H}}}\right). \tag{2.18}$$

In a nutshell, we assume that the covariance function of the GP k, is distributed according to a Wishart distribution. In other words, we have two nested processes, a Gaussian and a Wishart process, to model the data generation scheme. Hence we are studying a mixture of Gaussian processes. Note that the MAP2 (maximum a posteriori-2) method [MacKay, 1994] in Bayesian estimation leads to the same optimization problems as those arising from minimizing the regularized quality functional.

2.5 Hyperkernels

Having introduced the theoretical basis of the Hyper-RKHS, it is natural to ask whether $\underline{k}$ exist which satisfy the conditions of Definition 8. We address this question by giving a set of general recipes for building such kernels.

2.5.1 Power Series Construction

Suppose k is a kernel such that $k(x, x') \geq 0$ for all $x, x' \in \mathcal{X}$, and suppose $g : \mathbb{R} \to \mathbb{R}$ is a function with positive Taylor expansion coefficients $g(\xi) = \sum_{i=0}^{\infty} c_i \xi^i$ and convergence radius R. Then for pointwise positive $k(x, x') \leq \sqrt{R}$,

$$\underline{k}(\underline{x}, \underline{x}') := g(k(\underline{x})k(\underline{x}')) = \sum_{i=0}^{\infty} c_i (k(\underline{x})k(\underline{x}'))^i \tag{2.19}$$

is a hyperkernel. For $\underline{k}$ to be a hyperkernel, we need to check that firstly, $\underline{k}$ is a kernel, and secondly, for any fixed $\underline{x}$, $\underline{k}(\underline{x}, (x, x'))$ is a kernel. To see this, observe that for any fixed $\underline{x}$, $\underline{k}(\underline{x}, (x, x'))$ is a sum of kernel functions, hence it is a kernel itself (since $k^p(x, x')$ is a kernel if k is, for $p \in \mathbb{N}$). To show that $\underline{k}$ is a kernel, note that $\underline{k}(\underline{x}, \underline{x}') = \langle \underline{\Phi}(\underline{x}), \underline{\Phi}(\underline{x}') \rangle$, where $\underline{\Phi}(\underline{x}) := (\sqrt{c_0}, \sqrt{c_1} k^1(\underline{x}), \sqrt{c_2} k^2(\underline{x}), \ldots)$. Note that we require pointwise positivity, so that the coefficients of the sum in Equation (2.19) are always positive. The Gaussian RBF kernel satisfies this condition, but polynomial kernels of odd degree are not always pointwise positive. In the following example, we use the Gaussian kernel to construct a hyperkernel.

Example 7 (Harmonic Hyperkernel) *Suppose k a kernel with range $[0, 1]$, (RBF kernels satisfy this property), and set $c_i := (1 - \lambda_h)\lambda_h^i$, $i \in \mathbb{N}$, for some $0 < \lambda_h < 1$. Then we have*

$$\underline{k}(\underline{x}, \underline{x}') = (1 - \lambda_h) \sum_{i=0}^{\infty} \left(\lambda_h k(\underline{x}) k(\underline{x}') \right)^i = \frac{1 - \lambda_h}{1 - \lambda_h k(\underline{x}) k(\underline{x}')}. \tag{2.20}$$

For $k(x, x') = \exp(-\sigma^2 \|x - x'\|^2)$ this construction leads to

$$\underline{k}((x, x'), (x'', x''')) = \frac{1 - \lambda_h}{1 - \lambda_h \exp\left(-\sigma^2 (\|x - x'\|^2 + \|x'' - x'''\|^2)\right)}. \tag{2.21}$$

As one can see, for $\lambda_h \to 1$, $\underline{k}$ converges to $\delta_{\underline{x}, \underline{x}'}$, and thus $\|k\|^2_{\underline{\mathcal{H}}}$ converges to the Frobenius norm of k on $\mathcal{X} \times \mathcal{X}$.

It is straightforward to find other hyperkernels of this sort, simply by consulting tables on power series of functions. Table 2.2 contains a short list of suitable expansions. This hyperkernel can be thought of as a "parameter tuning" kernel, as it generates a sequence of kernel widths $\sigma^2, \sigma^4, \ldots$, which is generated by the power series expansion.

$g(\xi)$	Power series expansion	Radius of Convergence
$\exp \xi$	$1 + \frac{1}{1!}\xi + \frac{1}{2!}\xi^2 + \frac{1}{3!}\xi^3 + \ldots + \frac{1}{n!}\xi^n + \ldots$	∞
$\sinh \xi$	$\frac{1}{1!}\xi + \frac{1}{3!}\xi^3 + \frac{1}{5!}\xi^5 + \ldots + \frac{1}{(2n+1)!}\xi^{(2n+1)} + \ldots$	∞
$\cosh \xi$	$1 + \frac{1}{2!}\xi^2 + \frac{1}{4!}\xi^4 + \ldots + \frac{1}{(2n)!}\xi^{(2n)} + \ldots$	∞
$\mathrm{arctanh}\xi$	$\frac{\xi}{1} + \frac{\xi^3}{3} + \frac{\xi^5}{5} + \ldots + \frac{\xi^{2n+1}}{2n+1} + \ldots$	1
$-\ln(1 - \xi)$	$\frac{\xi}{1} + \frac{\xi^2}{2} + \frac{\xi^3}{3} + \ldots + \frac{\xi^n}{n} + \ldots$	1

Table 2.2: Hyperkernels by Power Series Construction.

However, if we want the kernel to adapt automatically to different widths for each dimension, we need to perform the summation that led to (2.20) for each dimension in its arguments separately. Such a hyperkernel corresponds to ideas developed in automatic relevance determination (ARD) [MacKay, 1994, Neal, 1996].

Example 8 (Hyperkernel for ARD) *Let $k_\Sigma(x,x') = \exp(-d_\Sigma(x,x'))$, where we define a distance by $d_\Sigma(x,x') = (x-x')^\top \Sigma(x-x')$, and Σ a diagonal covariance matrix. Take sums over each diagonal entry Σ_{ii} separately to obtain*

$$\begin{aligned} \underline{k}((x,x'),(x'',x''')) &= (1-\lambda_h) \sum_{diag(\sigma_1^{n_1},\ldots,\sigma_d^{n_d})} \sum_i \left(\lambda_h k_\Sigma(x,x') k_\Sigma(x'',x''')\right)^i \qquad (2.22) \\ &= \prod_{i=1}^d \frac{1-\lambda_h}{1-\lambda_h \exp\left(-\sigma_i((x_i-x_i')^2+(x_i''-x_i''')^2)\right)}. \end{aligned}$$

Here the sum goes over all $(\sigma_1^{n_1},\ldots,\sigma_d^{n_d})$ with $n_i \in \mathbb{N}, i=1,\ldots,d$. A similar definition also allows us to utilize a distance metric $d(x,x')$ which is a generalized radial distance as defined by Haussler [1999].

2.5.2 Hyperkernels Invariant to Translation

Another approach to constructing hyperkernels is to utilize an extension of a result due to Smola et al. [1998] concerning the Fourier transform of translation invariant kernels.

Theorem 12 (Translation Invariant Hyperkernel) *Suppose $\underline{k}((x_1-x_1'),(x_2-x_2'))$ is a function which depends on its arguments only via x_1-x_1' and x_2-x_2'. Let $\mathcal{F}_1\underline{k}(\omega,(x_2-x_2'))$ denote the Fourier transform with respect to (x_1-x_1'). If $\underline{k}(\tau,\tau') \geq 0$ for all τ,τ', and $\mathcal{F}_1\underline{k}(\omega,(x''-x''')) \geq 0$ for all $(x''-x''')$ and ω, then the function $\underline{k}$ is a hyperkernel.*

Proof From [Smola et al., 1998] we know that for $\underline{k}$ to be a kernel in one of its arguments, its Fourier transform has to be nonnegative. This yields the second condition. Next, we need to show that $\underline{k}$ is a kernel in its own right. Mercer's condition requires that for arbitrary f the following is positive:

$$\begin{aligned} & \int f(x_1,x_1')f(x_2,x_2')\underline{k}((x_1-x_1'),(x_2-x_2'))dx_1dx_1'dx_2dx_2' \\ = & \int f(\tau_1+x_1',x_1')f(\tau_2+x_2',x_2')dx_1'dx_2'\underline{k}(\tau_1,\tau_2)d\tau_1d\tau_2 \\ = & \int g(\tau_1)g(\tau_2)\underline{k}(\tau_1,\tau_2)d\tau_1d\tau_2 \end{aligned}$$

where $\tau_1 = x_1-x_1'$ and $\tau_2 = x_2-x_2'$. Here g is obtained by integration over x_1 and x_2 respectively. The latter is exactly Mercer's condition on $\underline{k}$, when viewed as a function of two variables only. ■

This means that we can check whether a radial basis function (e.g. Gaussian RBF, exponential RBF, damped harmonic oscillator, generalized B_n spline), can be used to construct a hyperkernel by checking whether their Fourier transform is positive.

2.5.3 Explicit Expansion

If we have a finite set of kernels that we want to choose from, this results in a hyperkernel which is a finite sum of possible kernel functions. This setting is similar that of Lanckriet et al. [2002].

Suppose $k_i(x, x')$ is a kernel for each $i = 1, \ldots, n$ (e.g. the RBF kernel or the polynomial kernel), then for $c_i \geqslant 0$,

$$\underline{k}(\underline{x}, \underline{x}') := \sum_{i=1}^{n} c_i k_i(\underline{x}) k_i(\underline{x}'), k_i(\underline{x}) \geqslant 0, \forall \underline{x} \tag{2.23}$$

is a hyperkernel, as can be seen by an argument similar to that of section 2.5.1. $\underline{k}$ is a kernel since $\underline{k}(\underline{x}, \underline{x}') = \langle \underline{\Phi}(\underline{x}), \underline{\Phi}(\underline{x}') \rangle$, where $\underline{\Phi}(\underline{x}) := (\sqrt{c_1} k_1(\underline{x}), \sqrt{c_2} k_2(\underline{x}), \ldots, \sqrt{c_n} k_n(\underline{x}))$.

Example 9 (Polynomial and RBF combination) *Let* $k_1(x, x') = (\langle x, x' \rangle + b)^{2p}$ *for some choice of* $b \in \mathbb{R}$ *and* $p \in \mathbb{N}$*, and* $k_2(x, x') = \exp(-\sigma^2 \|x - x'\|^2)$*. Then,*

$$\begin{aligned} \underline{k}((x_1, x_1'), (x_2, x_2')) &= c_1(\langle x_1, x_1' \rangle + b)^{2p}(\langle x_2, x_2' \rangle + b)^{2p} \\ &+ c_2 \exp(-\sigma^2 \|x_1 - x_1'\|^2) \exp(-\sigma^2 \|x_2 - x_2'\|^2) \end{aligned} \tag{2.24}$$

is a hyperkernel.

For a particular application, it is common to have several possible choices of representation and hence kernel functions. Even with fixed data representation, there are many possible choices of kernels for a particular data structure. In this setting, having an explicit expansion would allow the user to let the data determine the best kernel.

2.5.4 Summary of Hyper-RKHS

To learn the kernel, we use an approach similar to structural risk minimization. We achieve this by defining a quality functional which measures the performance of a given kernel, and a regularization term which is the squared norm of the kernel. This norm is defined by the inner product of the RKHS on the space of kernels itself. This inner product can be thought of as a generalised version of the inner product between two square matrices. The formulation gives rise to a hierarchical Bayesian interpretation, where the estimator is drawn from a Gaussian process which in turn is defined by the covariance matrix which is drawn from a Wishart process. We then proceed to show several examples of hyperkernels which can be used for estimation. In the following chapter, we express the regularised quality functional Q_{reg} as a convex optimization problem.

Chapter 3

Machine Learning with Hyperkernels

This chapter presents a semidefinite programming formulation for the problem of minimizing the regularized quality functional Q_{reg}, as defined in Chapter 2. In Section 3.1, we derive specific semidefinite programs (SDPs), using the approach of Lanckriet et al. [2004], for several common SVMs and Alignment. Experimental results for binary classification, regression and novelty detection are presented in Section 3.2.

3.1 Optimization Problem

We will now consider the optimization of the quality functionals utilizing hyperkernels. We choose the regularized risk functional as the empirical quality functional, that is we set $Q_{\text{emp}}(k, X, Y) := R_{\text{reg}}(f, X, Y)$. It is possible to utilize other quality functionals, such as the Alignment (Example 15). We focus our attention on the regularized risk functional, which is commonly used in SVMs. For a particular loss function $l(x_i, y_i, f(x_i))$, we obtain the regularized quality functional,

$$\min_{k \in \underline{\mathcal{H}}} \min_{f \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \ell(x_i, y_i, f(x_i)) + \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2 + \frac{\lambda_Q}{2} \|k\|_{\underline{\mathcal{H}}}^2, \tag{3.1}$$

where $\mathcal{H}$ is the RKHS associated with kernel k, and $\underline{\mathcal{H}}$ is the hyper-RKHS. By the representer theorem (Theorem 5 and Corollary 10) we can write the regularizers as quadratic terms,

$$\min_{\beta} \min_{\alpha} \frac{1}{m} \sum_{i=1}^{m} \ell(x_i, y_i, f(x_i)) + \frac{\lambda}{2} \alpha^\top K \alpha + \frac{\lambda_Q}{2} \beta^\top \underline{K} \beta \tag{3.2}$$

where $\alpha \in \mathbb{R}^m$ are the coefficients of the kernel expansion (Equation 1.3), and $\beta \in \mathbb{R}^{m^2}$ are the coefficients of the hyperkernel expansion (Equation 2.15). Using the approach in Lanckriet et al. [2004], the corresponding optimization problems for various loss functions can be expressed as SDPs. In general, solving a SDP would be take longer than solving a quadratic program (a traditional SVM is a quadratic program). This reflects the added cost incurred for optimizing over a class of kernels.

3.1.1 Semidefinite Programming Formulations

Semidefinite programming [Vandenberghe and Boyd, 1996] is the optimization of a linear objective function subject to constraints which are linear matrix inequalities and affine equalities.

Definition 13 (Semidefinite Program) *A semidefinite program (SDP) is a problem of the form:*

$$\begin{aligned} \min_{x} \quad & c^\top x \\ \text{subject to} \quad & F_0 + \sum_{i=1}^{q} x_i F_i \succeq 0 \textit{ and } Ax = b \end{aligned} \tag{3.3}$$

where $x \in \mathbb{R}^q$ are the decision variables, $A \in \mathbb{R}^{p \times q}$, $b \in \mathbb{R}^p$, $c \in \mathbb{R}^q$, and $F_i \in \mathbb{R}^{r \times r}$ are given. $X \succeq 0$ denotes that X is positive semidefinite.

In general, linear constraints $Ax + a \geqslant 0$ can be expressed as a semidefinite constraint $diag(Ax + a) \succeq 0$, and a convex quadratic constraint $(Ax + b)^\top (Ax + b) - c^\top x - d \leqslant 0$ can be written as

$$\begin{bmatrix} I & Ax + b \\ (Ax + b)^\top & c^\top x + d \end{bmatrix} \succeq 0,$$

using the Schur complement lemma (Theorem 39). When $t \in \mathbb{R}$, we can write the quadratic constraint $a^\top A a \leqslant t$ as $\|A^{\frac{1}{2}} a\|^2 \leqslant t$. In practice, linear and quadratic constraints are simpler and faster to implement in a convex solver and hence the above substitutions are rarely made.

We derive the corresponding SDP for Equation (3.1). The following proposition allows us to derive a SDP from a class of general convex programs. It follows the approach in Lanckriet et al. [2004], with some care taken with Schur complements of positive semidefinite matrices [Albert, 1969], and its proof can be found in Appendix A.1.

Proposition 14 (Quadratic Minimax) *Let $m, n, M \in \mathbb{N}$ and $H : \mathbb{R}^n \to \mathbb{R}^{m \times m}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ be linear maps. Let $A \in \mathbb{R}^{M \times m}$ and $a \in \mathbb{R}^M$. Also, let $d : \mathbb{R}^n \to \mathbb{R}$ and $G(\xi)$ be a function and the further constraints on ξ respectively. Then the optimization* *problem*

$$\begin{aligned} \min_{\xi \in \mathbb{R}^n} \max_{x \in \mathbb{R}^m} \quad & -\tfrac{1}{2}x^\top H(\xi)x - c(\xi)^\top x + d(\xi) \\ \textit{subject to} \quad & H(\xi) \succeq 0 \\ & Ax + a \geqslant 0 \\ & G(\xi) \succeq 0 \end{aligned} \tag{3.4}$$

can be rewritten as

$$\begin{aligned} \min_{t,\xi,\gamma} \quad & \tfrac{1}{2}t + a^\top \gamma + d(\xi) \\ \textit{subject to} \quad & \begin{bmatrix} \mathit{diag}(\gamma) & 0 & 0 & 0 \\ 0 & G(\xi) & 0 & 0 \\ 0 & 0 & H(\xi) & (A^\top\gamma - c(\xi)) \\ 0 & 0 & (A^\top\gamma - c(\xi))^\top & t \end{bmatrix} \succeq 0 \end{aligned} \tag{3.5}$$

in the sense that the ξ which solves (3.5) also solves (3.4).

Specifically, when we have the regularized quality functional, $d(\xi)$ is quadratic, and hence we obtain an optimization problem which has a mix of linear, quadratic and semidefinite constraints.

Corollary 15 *Let H, c, A and a be as in Proposition 14, and $\Sigma \succeq 0$. Then the solution ξ^* to the optimization problem*

$$\begin{aligned} \min_{\xi} \max_{x} \quad & -\tfrac{1}{2}x^\top H(\xi)x - c(\xi)^\top x + \tfrac{1}{2}\xi^\top \Sigma \xi \\ \textit{subject to} \quad & H(\xi) \succeq 0 \\ & Ax + a \geqslant 0 \\ & \xi \geqslant 0 \end{aligned} \tag{3.6}$$

can be found by solving the semidefinite programming problem

$$\begin{aligned} \min_{t,t',\xi,\gamma} \quad & \tfrac{1}{2}t + \tfrac{1}{2}t' + a^\top \gamma \\ \textit{subject to} \quad & \gamma \geqslant 0 \\ & \xi \geqslant 0 \\ & \|\Sigma^{\frac{1}{2}}\xi\|^2 \leqslant t' \\ & \begin{bmatrix} H(\xi) & (A^\top\gamma - c(\xi)) \\ (A^\top\gamma - c(\xi))^\top & t \end{bmatrix} \succeq 0 \end{aligned} \tag{3.7}$$

Proof By applying Proposition 14, and introducing an auxiliary variable t' which upper bounds the quadratic term of ξ, the claim is proved. ■

Comparing the objective function in Equation (3.6) with Equation (3.2), we observe that $H(\xi)$ and $c(\xi)$ are linear in ξ. Let $\xi' = \varepsilon\xi$. As we vary ε the constraints are still satisfied, but the objective function scales with ε. Since ξ is the coefficient in the hyperkernel expansion, this implies that we have a set of possible kernels which are just scalar multiples of each other. To avoid this, we add an additional constraint on ξ which is $\mathbf{1}^\top \xi = c$, where c is a constant. This breaks the scaling freedom of the kernel matrix. As a side-effect, the numerical stability of the SDP improves considerably. We chose a linear constraint so that it does not add too much overhead to the optimization problem.

We make one additional simplification of the optimization problem, which is to replace the upper bound of the squared norm ($\|\Sigma^{\frac{1}{2}}\xi\|^2 \leqslant t'$) with and upper bound on the norm ($\|\Sigma^{\frac{1}{2}}\xi\| \leqslant t'$).

3.1.2 Examples of Hyperkernel Optimization Problems

From the general framework above, we derive several examples of machine learning problems, specifically binary classification, regression, single class (also known as novelty detection) and alignment problems. The following examples illustrate our method for simultaneously optimizing over the class of kernels induced by the hyperkernel, as well as the hypothesis class of the machine learning problem. We consider machine learning problems based on kernel methods which are derived from Equation (3.1). The derivation is basically by application of Corollary 15, and are shown in Appendix A.2.

In this subsection, we define the following notation. For $p, q, r \in \mathbb{R}^n, n \in \mathbb{N}$ let $r = p \circ q$ be defined as element by element multiplication, $r_i = p_i \times q_i$. The pseudo-inverse (or Moore-Penrose inverse) of a matrix K is denoted $K^\dagger$. Let $\vec{K}$ be the m^2 by 1 vector formed by concatenating the columns of an m by m matrix. We define the hyperkernel Gram matrix $\underline{K}$ by putting together m^2 of these vectors, that is we set $\underline{K} = [\vec{K}_{pq}]_{p,q=1}^m$. Other notations include: the kernel matrix $K = \text{reshape}(\underline{K}\beta)$ (reshaping a m^2 by 1 vector, $\underline{K}\beta$, to a m by m matrix), $Y = \text{diag}(y)$ (a matrix with y on the diagonal and zero everywhere else), $G(\beta) = YKY$ (the dependence on β is made explicit), $\mathbf{I}$ the identity matrix, $\mathbf{1}$ a vector of ones and $\mathbf{1}_{m\times m}$ a matrix of ones. Let w be the weight vector and b_{offset} the bias term in feature space, that is the hypothesis function in feature space is defined as $g(x) = w^\top \phi(x) + b_{\mathit{offset}}$ where $\phi(\cdot)$ is the feature mapping defined by the kernel function k.

The number of training examples is assumed to be m, that is $X_{\text{train}} = \{x_1, \ldots, x_m\}$ and $Y_{\text{train}} = y = \{y_1, \ldots, y_m\}$. Where appropriate, γ and χ are Lagrange multipliers, while η and ξ are vectors of Lagrange multipliers from the derivation of the Wolfe dual for the SDP, β are the hyperkernel coefficients, t_1 and t_2 are the auxiliary variables.

When $\eta \in \mathbb{R}^m$, we define $\eta \geqslant 0$ to mean that each $\eta_i \geqslant 0$ for $i = 1, \dots, m$.

Example 10 (L_1 SVM (C-parameterization)) *A commonly used support vector classifier, the C-SVM [Bennett and Mangasarian, 1992, Cortes and Vapnik, 1995] uses an L_1 soft margin, $l(x_i, y_i, f(x_i)) = \max(0, 1 - y_i f(x_i))$, which allows errors on the training set. The parameter C is given by the user. Setting the quality functional $Q_{\mathrm{emp}}(k, X, Y) = \min_{f \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^m l(x_i, y_i, f(x_i)) + \frac{1}{2C} \|w\|^2_{\mathcal{H}}$, the resulting SDP is*

$$\begin{array}{ll} \min\limits_{\beta,\gamma,\eta,\xi} & \frac{1}{2}t_1 + \frac{C}{m}\xi^\top \mathbf{1} + \frac{\lambda_Q}{2}t_2 \\ \textit{subject to} & \eta \geqslant 0, \xi \geqslant 0, \beta \geqslant 0 \\ & \|\underline{K}^{\frac{1}{2}}\beta\| \leqslant t_2, \mathbf{1}^\top \beta = 1 \\ & \begin{bmatrix} G(\beta) & z \\ z^\top & t_1 \end{bmatrix} \succeq 0, \end{array} \tag{3.8}$$

where $z = \gamma y + \mathbf{1} + \eta - \xi$.

The value of the support vector coefficients, α, which optimizes the corresponding Lagrange function is $G(\beta)^\dagger z$, and the classification function, $f = sign(K(\alpha \circ y) - b_{offset})$, is given by $f = sign(KG(\beta)^\dagger (y \circ z) - \gamma)$.

Example 11 (Linear SVM (ν-style)) *An alternative parameterization of the ℓ_1 soft margin was introduced by Schölkopf et al. [2000], where the user defined parameter $\nu \in [0, 1]$ controls the fraction of margin errors and support vectors. Using ν-SVM as Q_{emp}, that is, for a given ν, $Q_{\mathrm{emp}}(k, X, Y) = \min_{f \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^m \zeta_i + \frac{1}{2}\|w\|^2_{\mathcal{H}} - \nu\rho$ subject to $y_i f(x_i) \geqslant \rho - \zeta_i$ and $\zeta_i \geqslant 0$ for all $i = 1, \dots, m$. The corresponding SDP is given by*

$$\begin{array}{ll} \min\limits_{\beta,\gamma,\eta,\xi,\chi} & \frac{1}{2}t_1 - \chi\nu + \xi^\top \frac{1}{m} + \frac{\lambda_Q}{2}t_2 \\ \textit{subject to} & \chi \geqslant 0, \eta \geqslant 0, \xi \geqslant 0, \beta \geqslant 0 \\ & \|\underline{K}^{\frac{1}{2}}\beta\| \leqslant t_2, \mathbf{1}^\top \beta = 1 \\ & \begin{bmatrix} G(\beta) & z \\ z^\top & t_1 \end{bmatrix} \succeq 0 \end{array} \tag{3.9}$$

where $z = \gamma y + \chi \mathbf{1} + \eta - \xi$.

The value of the support vector coefficients, α, which optimizes the corresponding Lagrange function is $G(\beta)^\dagger z$, and the classification function, $f = sign(K(\alpha \circ y) - b_{offset})$, is given by $f = sign(KG(\beta)^\dagger (y \circ z) - \gamma)$.

Example 12 (Quadratic SVM) *Instead of using an ℓ_1 loss class, Mangasarian and Musicant [2001] uses an ℓ_2 loss class,*

$$l(x_i, y_i, f(x_i)) = \begin{cases} 0 & \textit{if } y_i f(x_i) \geqslant 1 \\ (1 - y_i f(x_i))^2 & \textit{otherwise} \end{cases},$$

and regularized the weight vector as well as the bias term, that is the empirical quality functional is set to $Q_{\mathrm{emp}}(k, X, Y) = \min_{f \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \zeta_i^2 + \frac{1}{2}(\|w\|_{\mathcal{H}}^2 + b_{offset}^2)$ subject to $y_i f(x_i) \geqslant 1 - \zeta_i$ and $\zeta_i \geqslant 0$ for all $i = 1, \ldots, m$. This is also known as the Lagrangian SVM. The resulting dual SVM problem has fewer constraints, as is evidenced by the smaller number of Lagrange multipliers needed in the SDP below.

$$\begin{array}{ll} \min\limits_{\beta,\eta} & \frac{1}{2}t_1 + \frac{\lambda_Q}{2}t_2 \\ \textit{subject to} & \eta \geqslant 0, \beta \geqslant 0 \\ & \|\underline{K}^{\frac{1}{2}}\beta\| \leqslant t_2, \mathbf{1}^\top \beta = 1 \\ & \begin{bmatrix} H(\beta) & (\eta + \mathbf{1}) \\ (\eta + \mathbf{1})^\top & t_1 \end{bmatrix} \succeq 0 \end{array} \tag{3.10}$$

where $H(\beta) = Y(K + \mathbf{1}_{m \times m} + \lambda m \mathbf{I})Y$, and $z = \gamma \mathbf{1} + \eta - \xi$.

The value of the support vector coefficients, α, which optimizes the corresponding Lagrange function is $H(\beta)^\dagger(\eta + \mathbf{1})$, and the classification function, $f = sign(K(\alpha \circ y) - b_{offset})$, is given by $f = sign(KH(\beta)^\dagger((\eta + \mathbf{1}) \circ y) + y^\top(H(\beta)^\dagger(\eta + \mathbf{1})))$.

Example 13 (Single class SVM) *For unsupervised learning, the single class SVM computes a function which captures regions in input space where the probability density is in some sense large Schölkopf et al. [2001]. The quality functional $Q_{\mathrm{emp}}(k, X, Y) = \min_{f \in \mathcal{H}} \frac{1}{\nu m} \sum_{i=1}^{m} \zeta_i + \frac{1}{2}\|w\|_{\mathcal{H}}^2 - \rho$ subject to $f(x_i) \geqslant \rho - \zeta_i$, and $\zeta_i \geqslant 0$ for all $i = 1, \ldots, m$, and $\rho \geqslant 0$. The corresponding SDP for this problem, also known as novelty detection, is shown below.*

$$\begin{array}{ll} \min\limits_{\beta,\gamma,\eta,\xi} & \frac{1}{2}t_1 + \xi^\top \frac{1}{\nu m} - \gamma + \frac{\lambda_Q}{2\nu}t_2 \\ \textit{subject to} & \eta \geqslant 0, \xi \geqslant 0, \beta \geqslant 0 \\ & \|\underline{K}^{\frac{1}{2}}\beta\| \leqslant t_2, \mathbf{1}^\top \beta = 1 \\ & \begin{bmatrix} K & z \\ z^\top & t_1 \end{bmatrix} \succeq 0 \end{array} \tag{3.11}$$

where $z = \gamma \mathbf{1} + \eta - \xi$, and $\nu \in [0, 1]$ a user selected parameter controlling the proportion of the data to be classified as novel.

The score to be used for novelty detection is given by $f = K\alpha - b_{offset}$, which reduces to $f = \eta - \xi$, by substituting $\alpha = K^\dagger(\gamma \mathbf{1} + \eta - \xi)$, $b_{offset} = \gamma \mathbf{1}$ and $K = reshape(\underline{K}\beta)$.

Example 14 (ν-Regression) *We derive the SDP for ν regression [Schölkopf et al., 2000], which automatically selects the ε insensitive tube for regression. As in the ν-SVM case in Example 11, the user defined parameter ν controls the fraction of errors and support vectors. Using the ε-insensitive loss, $l(x_i, y_i, f(x_i)) = \max(0, |y_i - f(x_i)| - \varepsilon)$, and the ν-parameterized quality functional, $Q_{\mathrm{emp}}(k, X, Y) = \min_{f \in \mathcal{H}} C\left(\nu\varepsilon + \frac{1}{m}\sum_{i=1}^{m}(\zeta_i + \zeta_i^*)\right)$ subject to $f(x_i) - y_i \leqslant \varepsilon - \zeta_i$, $y_i - f(x_i) \leqslant \varepsilon - \zeta_i^*$, $\zeta_i^{(*)} \geqslant 0$ for all $i = 1, \ldots, m$ and*

$\varepsilon \geqslant 0$. *The corresponding SDP is*

$$\begin{array}{ll} \min\limits_{\beta,\gamma,\eta,\xi,\chi} & \frac{1}{2}t_1 + \frac{\chi\nu}{\lambda} + \xi^\top \frac{1}{m\lambda} + \frac{\lambda_Q}{2\lambda}t_2 \\ \textit{subject to} & \chi \geqslant 0, \eta \geqslant 0, \xi \geqslant 0, \beta \geqslant 0 \\ & \|\underline{K}^{\frac{1}{2}}\beta\| \leqslant t_2, \mathbf{1}^\top\beta = \mathrm{stddev}(\mathrm{Y}_{\mathrm{train}}) \\ & \begin{bmatrix} F(\beta) & z \\ z^\top & t_1 \end{bmatrix} \succeq 0 \end{array}, \tag{3.12}$$

where $z = \begin{bmatrix} -y \\ y \end{bmatrix} - \gamma \begin{bmatrix} \mathbf{1} \\ -\mathbf{1} \end{bmatrix} + \eta - \xi - \chi \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}$ *and* $F(\beta) = \begin{bmatrix} K & -K \\ -K & K \end{bmatrix}$.

The Lagrange function is minimized for $\alpha = F(\beta)^\dagger z$, *and substituting into* $f = K\alpha - b_{\mathit{offset}}$, *we obtain the regression function* $f = \begin{bmatrix} -K & K \end{bmatrix} F(\beta)^\dagger z - \gamma$.

Example 15 (Kernel Target Alignment) *For the Kernel Target Alignment approach [Cristianini et al., 2001],* $Q_{\mathrm{emp}} = y^\top K y$, *we directly minimize the regularized quality functional, obtaining the following optimization problem,*

$$\begin{array}{ll} \min\limits_{k\in\underline{\mathcal{H}}} & -\frac{1}{2}y^\top K y + \frac{\lambda_Q}{2}\beta^\top\underline{K}\beta \\ \textit{subject to} & \beta \geqslant 0 \end{array}$$

which can be expressed as:

$$\begin{array}{ll} \min\limits_{\beta} & \frac{1}{2}t_1 + \frac{\lambda_Q}{2}t_2 \\ \textit{subject to} & \beta \geqslant 0 \\ & \|\underline{K}^{\frac{1}{2}}\beta\|^2 \leqslant t_2 \\ & \begin{bmatrix} K & y \\ y^\top & t_1 \end{bmatrix} \succeq 0 \end{array} \tag{3.13}$$

Note that for the case of Alignment, Q_{emp} *does not provide a direct formulation for the hypothesis function, but instead, it determines a kernel matrix* K. *This kernel matrix,* K, *can be utilized in a traditional SVM, to obtain a classification function.*

3.2 Experiments

In the following experiments, we use data from the UCI repository [Blake and Merz, 1998]. Where the data attributes are numerical, we *do not perform any preprocessing* of the data. Boolean attributes are converted to $\{-1, 1\}$, and categorical attributes are arbitrarily assigned an order, and numbered $\{1, 2, \ldots\}$. The SDPs were solved using

SeDuMi [Sturm, 1999], and YALMIP [Löfberg, 2002] was used to convert the equations into standard form. We used the hyperkernel for automatic relevance determination (Equation 2.22) for the hyperkernel optimization problems. The scaling freedom that this hyperkernel provides for each dimension means we do not have to normalize data to some arbitrary distribution.

For the classification and regression experiments, the datasets were split into 100 random permutations of 60% training data and 40% test data. We deliberately did not attempt to tune parameters and instead made the following choices uniformly for all datasets in classification, regression and novelty detection:

- The kernel width σ_i, for each dimension, was set to 50 times the 90% quantile of the value of $|x_i - x_j|$ over all the training data. This ensures sufficient coverage without having too wide a kernel. This value was estimated from a random sampling of the training data.

- λ was adjusted so that $\frac{1}{\lambda m} = 100$ (that is $C = 100$ in the Vapnik-style parameterization of SVMs). This has commonly been reported to yield good results.

- $\nu = 0.3$. While this is clearly suboptimal for many datasets, we decided to choose it beforehand to avoid having to change *any* parameter. Clearly we could use previous reports on generalization performance to set ν to this value for better performance. For novelty detection, $\nu = 0.1$ (see Section 3.2.5 for details).

- λ_h for the Harmonic Hyperkernel was chosen to be 0.6, giving adequate coverage over various kernel widths in (2.20) (small λ_h focus almost exclusively on wide kernels, λ_h close to 1 will treat all widths equally).

- The hyperkernel regularization constant was set to $\lambda_Q = 1$.

- For the scale breaking constraint $\mathbf{1}^\top \beta = c$, c was set to 1 for classification as the hypothesis class only utilizes the sign of the trained function, and therefore is scale free. However, for regression, c is set to the standard deviation of Y_{train}, so that the hyperkernel coefficients are of the same scale as the output (the constant offset b_{offset} takes care of the mean).

Note that the SDP provides the coefficients of the linear combination of kernels β_{ij}, as well as the coefficients of the support vectors α_i. In the following experiments, these coefficients were used to compute the hypothesis function. Therefore, there is no need for an additional optimization step for finding the best hypothesis.

3.2.1 Low Rank Approximation

Although the optimization of Equation (3.1) has reduced the problem of optimizing over two possibly infinite dimensional Hilbert spaces to a finite problem, it is still formidable in practice as there are m^2 coefficients for β. For an explicit expansion of type in Equation (2.23) one can optimize in the expansion coefficients $k_i(\underline{x})k_i(\underline{x}')$ directly, which leads to a quality functional with an ℓ_2 penalty on the expansion coefficients. Such an approach is appropriate if there are few terms in (2.23).

In the general case (or if the explicit expansion has many terms), one can utilize a low-rank approximation, as described by Fine and Scheinberg [2001]. This entails picking from $\{\underline{k}((x_i, x_j), \cdot)|1 \leq i, j \leq m\}$ a small fraction of terms, p (where $m^2 \gg p$), which approximate $\underline{k}$ on $X_{\text{train}} \times X_{\text{train}}$ sufficiently well. In particular, we choose an $m \times p$ truncated lower triangular matrix G such that $\|P\underline{k}P^\top - GG^\top\|_F \leqslant \eta$, where P is the permutation matrix which sorts the eigenvalues of $\underline{k}$ into decreasing order, and η is the level of approximation needed. The norm, $\|\cdot\|_F$ is the Frobenius norm. In the following experiments, the hyperkernel matrix was approximated to $\eta = 10^{-6}$ using the incomplete Cholesky factorization method [Bach and Jordan, 2002].

3.2.2 Classification Experiments

Several binary classification datasets[1] from the UCI repository were used for the experiments. A set of synthetic data (labeled syndata in the results) sampled from two Gaussians was created to illustrate the scaling freedom between dimensions. The first dimension had a standard deviation of 1000 whereas the second dimension had a standard deviation of 1 (a sample result is shown in Figure 2.1). The results of the experiments are shown in Table 3.1.

From Table 3.1, we observe that our method achieves state of the art results for all the datasets, except the "heart" dataset. We also achieve results much better than previously reported for the "credit" dataset. Comparing the results for C-SVM and Tuned SVM, we observe that our method is always equally good, or better than a C-SVM tuned using 10-fold cross validation.

In an attempt to lower the computational time required for the experiments, we also performed low rank approximation of the kernel matrix, which effectively is a selection of a subset of the training data. Table 3.2 shows the results obtained when we approximate the *kernel matrix* using a tolerance of 10^{-6}. The number of data points selected was forced to be between 80 and 300, to control the size of the kernel matrix. The rightmost two columns are repeated from Table 3.1. The results from the approximate problem were all within one standard deviation of the method using

[1] We classified window vs. non-window for glass data, the other datasets are all binary.

Data	C-SVM	ν-SVM	Lag-SVM	Best other	CV Tuned SVM (C)
syndata	2.8±2.4	**1.9±1.9**	2.4±2.2	NA	5.9±5.4 (10^8)
pima	**23.5±2.0**	27.7±2.1	23.6±1.9	23.5	24.1±2.1 (10^4)
ionosph	6.6±1.8	6.7±1.8	6.4±1.9	**5.8**	6.1±1.8 (10^3)
wdbc	3.3±1.2	3.8±1.2	**3.0±1.1**	3.2	5.2±1.4 (10^6)
heart	19.7±3.3	19.3±2.4	20.1±2.8	**16.0**	23.2±3.7 (10^4)
thyroid	7.2±3.2	10.1±4.0	6.2±3.1	**4.4**	5.2±2.2 (10^5)
sonar	14.8±3.7	15.3±3.7	**14.7±3.6**	15.4	15.3±4.1 (10^3)
credit	14.6±1.8	**13.7±1.5**	14.7±1.8	22.8	15.3±2.0 (10^8)
glass	6.0±2.4	8.9±2.6	**6.0±2.2**	NA	7.2±2.7 (10^3)

Table 3.1: Hyperkernel classification: Test error and standard deviation in percent. The second, third and fourth columns show the results of the hyperkernel optimizations of C-SVM (Example 10), ν-SVM (Example 11) and Lagrangian SVM (Example 12) respectively. The results in the fifth column shows the best results from Freund and Schapire [1996], Rätsch et al. [2001] and Meyer et al. [2003]. The rightmost column shows a C-SVM tuned in the traditional way. A Gaussian RBF kernel was tuned using 10-fold cross validation on the training data, with the best value of C shown in brackets. A grid search was performed on (C, σ). The values of C tested were $\{10^{-2}, 10^{-1}, \dots, 10^9\}$. The values of the kernel width, σ, tested were between 10% and 90% quantile of the distance between a pair of sample of points in the data. These quantiles were estimated by a random sample of 20% of the training data.

all the data points. This shows the potential of using a subset of the training data to obtain an equally good classification result. The second column shows the average value of the constant η used in the approximation.

Data	Approx. η	C-SVM	ν-SVM	Lag-SVM	other	CV SVM (C)
syndata	0	2.9±2.4	1.9±1.9	2.4±2.1	NA	5.9±5.4 (10^8)
pima	4×10^{-7}	23.8±2.0	27.2±2.3	24.1±1.9	23.5	24.1±2.1 (10^4)
ionosph	5×10^{-7}	6.6±2.0	6.8±1.8	6.4±1.9	5.8	6.1±1.8 (10^3)
wdbc	3×10^{-4}	3.3±1.2	3.8±1.2	3.0±1.1	3.2	5.2±1.4 (10^6)
heart	2×10^{-7}	19.5±3.3	19.4±2.5	20.1±2.8	16.0	23.2±3.7 (10^4)
thyroid	1×10^{-9}	6.0±3.1	7.2±3.6	5.5±2.6	4.4	5.2±2.2 (10^5)
sonar	3×10^{-15}	14.8±3.7	15.6±3.8	14.8±3.5	15.4	15.3±4.1 (10^3)
credit	1×10^{-4}	14.8±1.8	13.8±1.6	14.8±1.8	22.8	15.3±2.0 (10^8)
glass	3×10^{-7}	5.9±2.4	8.5±2.6	5.8±2.2	NA	7.2±2.7 (10^3)

Table 3.2: Approximate kernel classification: Test error and standard deviation in percent

Ong et al. [2003] reported results which were based on an optimization problem where we iteratively alternated between optimizing the kernel coefficients and hyperkernel coefficients. This could potentially result in a local minima. In that setting, the regularized quality functional performed poorly on the Ionosphere dataset. This is not the case here, where we optimize using a SDP. Note that the results here for the tuned C-SVM (rightmost column) for the synthetic data and the credit data have improved over our earlier results (see [Ong and Smola, 2003]). This was because we only searched up till $C = 10^6$ in our earlier work, and we searched further for our current results. This demonstrates another advantage of the hyperkernel optimization over parameter selection using cross validation, we can only search a finite number of values (usually only a small number of these) for each parameter. Since the publication of this work in Ong et al. [2003] and Ong and Smola [2003], there have been improvements in the optimization problem from the semidefinite programs derived in Section 3.1.2 to second order cone programs [Tsang and Kwok, 2004].

3.2.3 Effect of λ_Q and λ_h on Classification Error

In order to investigate the effect of varying the hyperkernel regularization constant, λ_Q, and the Harmonic Hyperkernel parameter, λ_h, we performed experiments using the C-SVM hyperkernel optimization (Example 10). We performed two sets of experiments with each of our chosen datasets. The results shown in Table 3.3.

	λ_h		λ_Q	
Data	Error	Deviation	Error	Deviation
syndata	3.0±1.1	2.2	2.8±0.0	2.2
pima	25.7±2.6	1.9	24.5±0.1	1.5
ionosph	6.6±1.0	1.7	7.2±0.1	1.9
wdbc	2.9±0.4	0.9	2.7±0.2	0.8
heart	19.7±2.0	3.0	19.4±0.9	2.8
thyroid	6.5±2.8	3.0	6.7±0.3	3.7
sonar	15.7±1.6	3.4	15.1±0.2	3.3
credit	16.0±1.8	1.6	14.7±0.4	1.6
glass	5.9±1.0	2.3	5.2±0.3	2.3

Table 3.3: Effect of varying λ_h and λ_Q on classification error. In the left experiment, we fixed $\lambda_Q = 1$, and λ_h was varied with the values $\lambda_h = \{0.1, 0.2, \ldots, 0.9, 0.92, 0.94, 0.96, 0.98\}$.In the right, we set $\lambda_h = 0.6$ and varied $\lambda_Q = \{10^{-4}, 10^{-3}, \ldots, 10^5\}$. The error columns (columns 2 and 4) report the average error on the test set and the standard deviation of the error over the different parameter settings. The deviation columns (columns 3 and 5) report the average standard deviation over 10 random 60%/40% splits.

From Table 3.3 and Figure 3.1, we observe that the variation in classification accuracy over the whole range of the hyperkernel regularization constant, λ_Q is less than the standard deviation of the classification accuracies of the various datasets (compare with Table 3.1). This demonstrates that our method is quite insensitive to the regularization parameter over the range of values tested for the various datasets.

The method shows a higher sensitivity to the harmonic hyperkernel parameter, λ_h (Figure 3.2). Since this parameter effectively selects the scale of the problem, by selecting the "width" of the kernel, it is to be expected that each dataset would have a different ideal value of λ_h. It is to be noted that the generalization accuracy at $\lambda_h = 0.6$ is within one standard deviation (see Table 3.1) of the best accuracy achieved over the whole range tested.

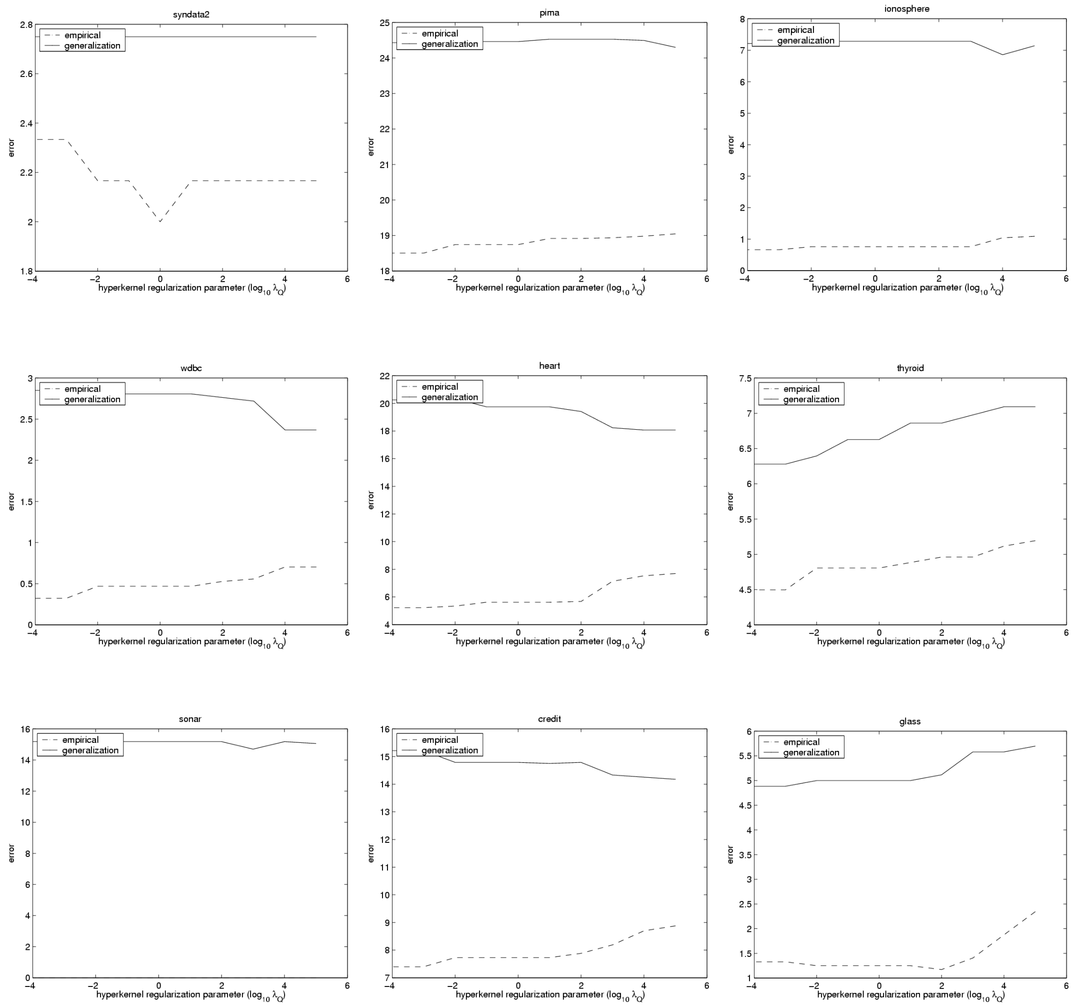


Figure 3.1: Effect of varying λ_Q on classification error. We set $\lambda_h = 0.6$ and varied $\lambda_Q = \{10^{-4}, 10^{-3}, \ldots, 10^{5}\}$. The empirical error of the Sonar dataset is zero for all values of λ_Q.

3.2.4 Regression Experiments

To demonstrate that we can solve problems other than binary classification using the same framework, we performed regression and novelty detection. The results of regression are shown in Table 3.4. We utilized *the same parameter settings* as in the previous section. The second column shows the results from the hyperkernel optimiza-

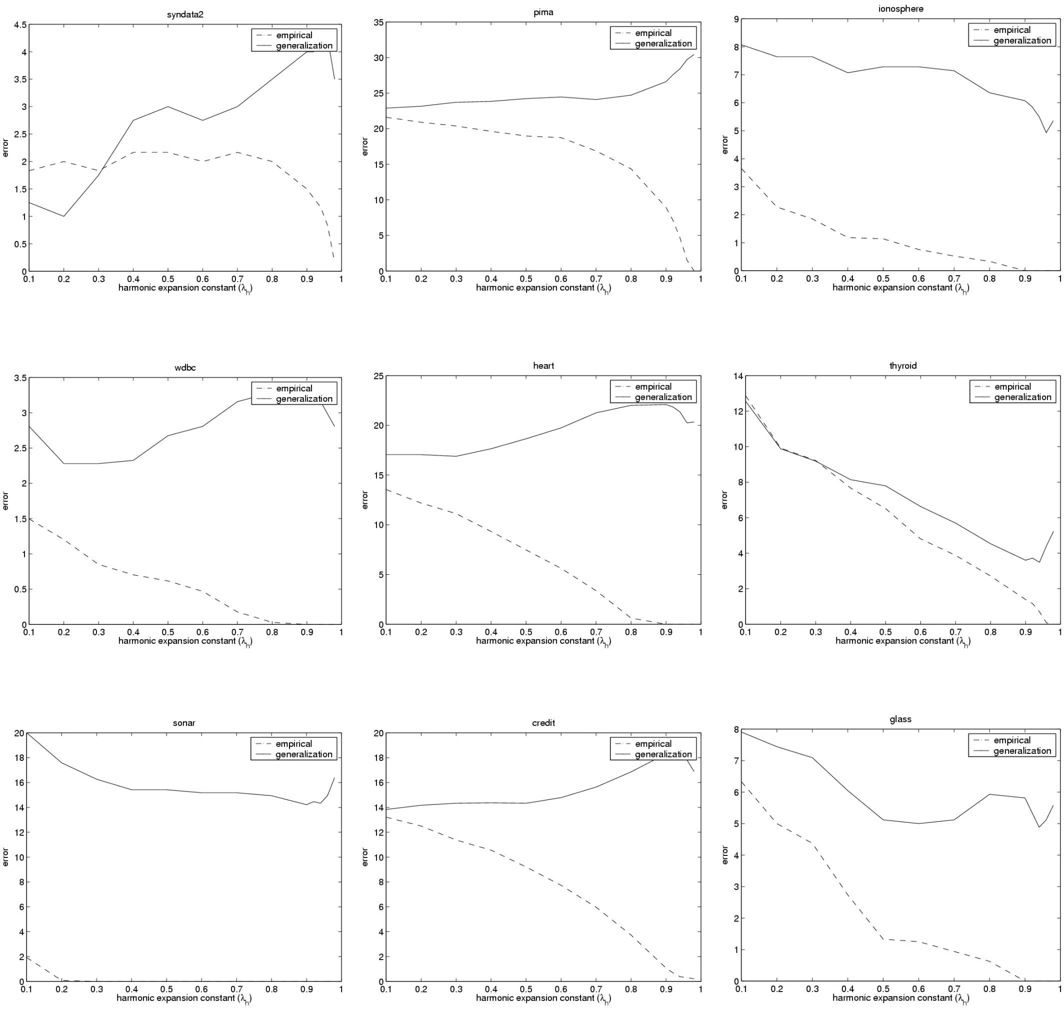


Figure 3.2: Effect of varying λ_h on classification error. We fixed $\lambda_Q = 1$, and λ_h was varied with the values $\lambda_h = \{0.1, 0.2, \ldots, 0.9, 0.92, 0.94, 0.96, 0.98\}$.

tion of the ν-regression (Example 14). The results in the third column shows the best results from [Meyer et al., 2003]. The rightmost column shows a ε-SVR with a Gaussian kernel tuned using 10-fold cross validation on the training data. Similar to the classification setting, grid search was performed on (C, σ). The values of C tested were $\{10^{-2}, 10^{-1}, \ldots, 10^{9}\}$. The values of the kernel width, σ, tested were between 10% and 90% quantile of the distance between a pair of sample of points in the data. These quantiles were estimated by a random sample of the training data.

Data	ν-SVR	Best other	Tuned ε-SVM
auto-mpg	7.76±1.05	7.11	9.61±1.18
boston	12.02±2.25	9.60	15.04±3.31
auto imports($\times 10^6$)	4.42±1.04	0.25	5.76±1.28
cpu($\times 10^3$)	4.25±2.89	3.16	9.79±7.29
servo	0.79±0.16	0.25	0.57±0.14

Table 3.4: Hyperkernel regression: Mean Squared Error

Meyer et al. [2003] used a 90%/10% split of the data for their experiments, while we used a 60%/40% split, which may account for the better performance in the cpu and servo datasets. The reason for the much better rate reported on the "auto imports" dataset remains a mystery.

3.2.5 Novelty Detection

We apply the single class support vector machine to detect outliers in the USPS data. The test set of the default split in the USPS database was used in the following experiments. The parameter ν was set to 0.1 for these experiments, hence selecting up to 10% of the data as outliers. Since there is no quantitative method for measuring the performance of novelty detection, we cannot directly compare our results with the traditional single class SVM. We can only subjectively conclude, by visually inspecting a sample of the digits, that our approach works for novelty detection of USPS digits. Figures 3.3 to 3.11 shows a sample of the digits 1 to 9 respectively. The top rows shows the digits identified as 'novel', and the bottom row shows 'common' digits.

3.3 Summary

We have shown that when the empirical quality functional is the regularized risk functional, the resulting optimization problem is convex. We derive several examples of common estimators for classification, regression and novelty detection. Since we can optimize over the whole class of kernel functions, we can define more general kernels which may have many free parameters, without overfitting. The experimental results

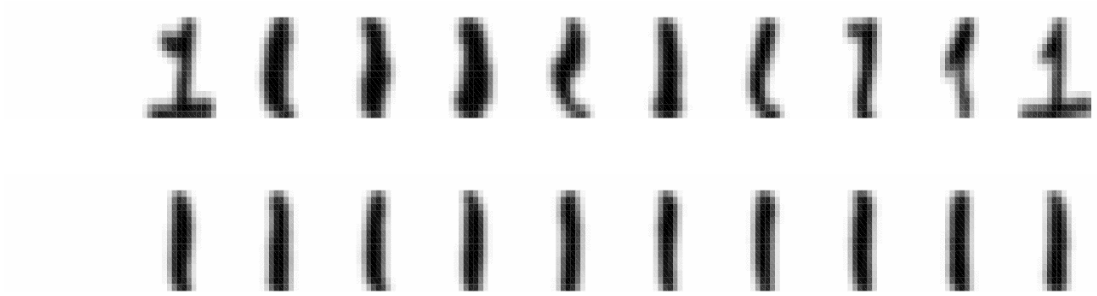


Figure 3.3: Top: Images of digit ‘1’ considered novel by algorithm; Bottom: Common images of digit ‘1’

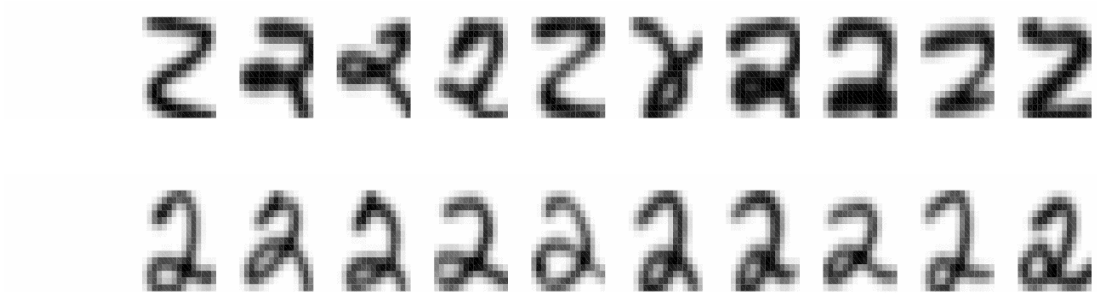


Figure 3.4: Top: Images of digit ‘2’ considered novel by algorithm; Bottom: Common images of digit ‘2’

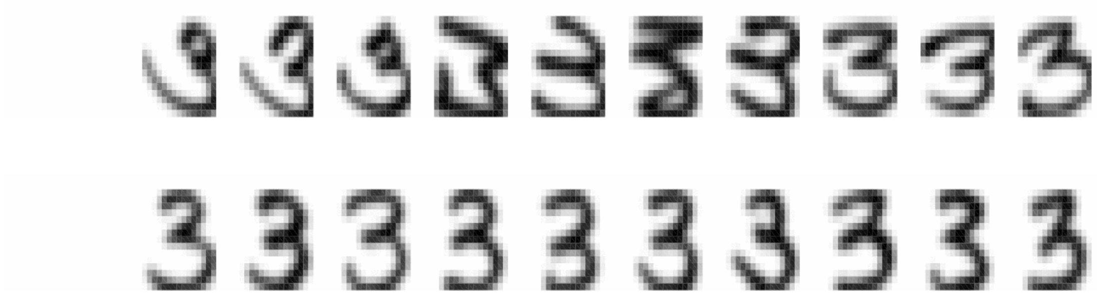


Figure 3.5: Top: Images of digit ‘3’ considered novel by algorithm; Bottom: Common images of digit ‘3’

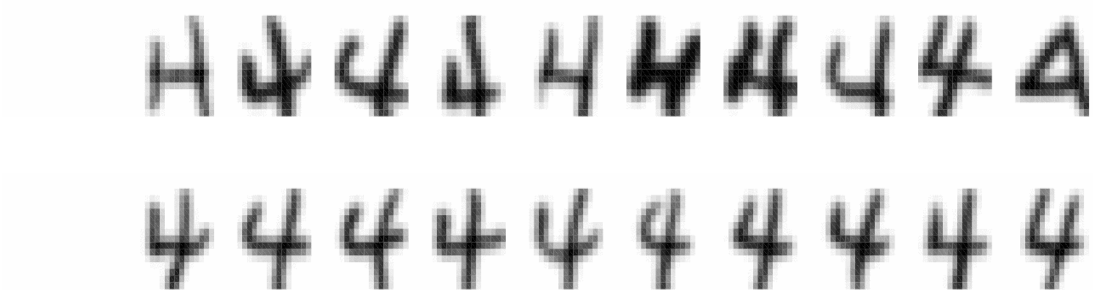


Figure 3.6: Top: Images of digit ‘4’ considered novel by algorithm; Bottom: Common images of digit ‘4’

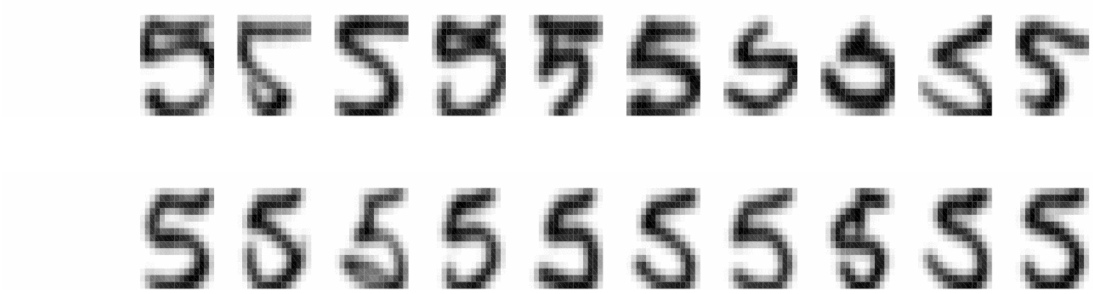


Figure 3.7: Top: Images of digit ‘5’ considered novel by algorithm; Bottom: Common images of digit ‘5’

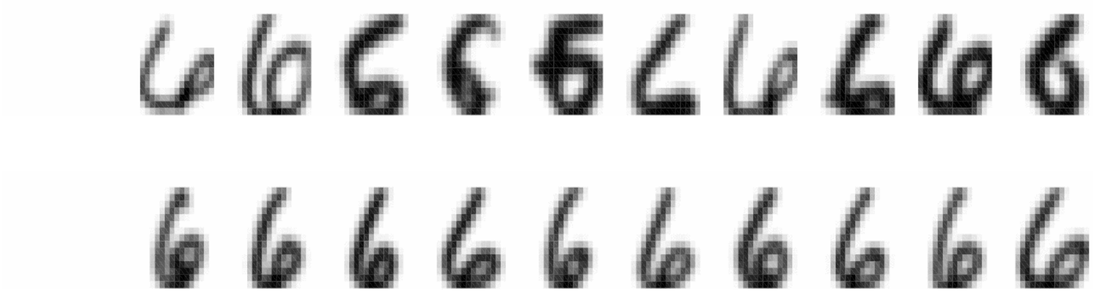


Figure 3.8: Top: Images of digit ‘6’ considered novel by algorithm; Bottom: Common images of digit ‘6’

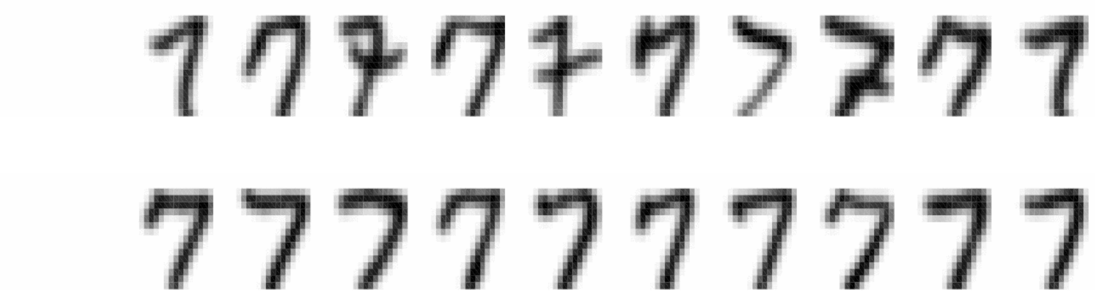


Figure 3.9: Top: Images of digit '7' considered novel by algorithm; Bottom: Common images of digit '7'

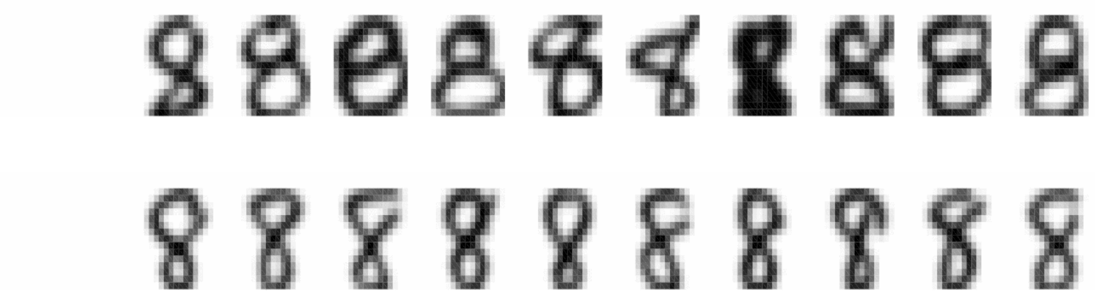


Figure 3.10: Top: Images of digit '8' considered novel by algorithm; Bottom: Common images of digit '8'

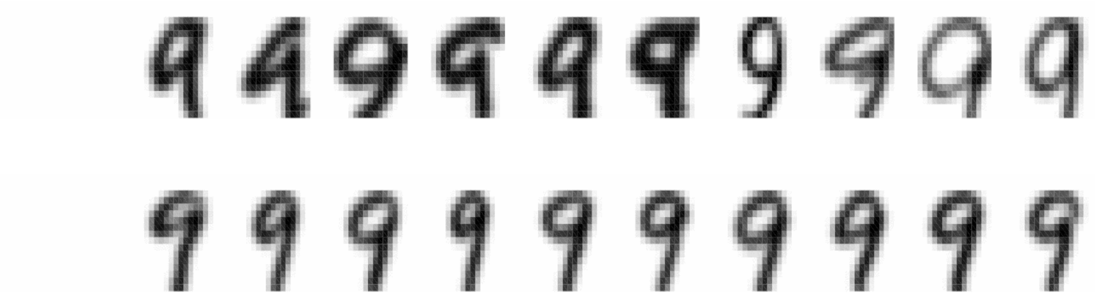


Figure 3.11: Top: Images of digit '9' considered novel by algorithm; Bottom: Common images of digit '9'

on classification demonstrate that it is possible to achieve the state of the art. Furthermore, the same framework and parameter settings work for various datasets as well as regression and novelty detection.

It is important to stress that our approach has made support vector estimation more automated. Parameter adjustment is less critical compared to the case when the kernel is fixed.

Chapter 4

Learning with Indefinite Kernels

In this chapter we show that kernel methods can be adapted to deal with indefinite kernels, that is, kernels which are not positive semidefinite. They do not satisfy Mercer's condition and they induce associated functional spaces called reproducing kernel Kreĭn spaces (RKKS) (Section 4.2), a generalization of reproducing kernel Hilbert spaces (RKHS).

Machine learning in RKKS shares many "nice" properties of learning in RKHS, such as orthogonality and projection. However, since the kernels are indefinite, we can no longer minimize the loss, instead we stabilize it (Section 4.3). We show a general representer theorem for constrained stabilization and prove generalization bounds by computing the Rademacher averages of the kernel class (Section 4.4). We list several examples of indefinite kernels and show some results on the spectrum of the operators useful for machine learning (Section 4.5).

4.1 Why Non-Positive Kernels?

Almost all current research on kernel methods in machine learning focuses on functions $k(x, x')$ which are positive semidefinite. That is, it focuses on kernels which satisfy Mercer's condition and which consequently can be seen as scalar products in some Hilbert space.

In Chapter 2 of this thesis, the proposed approach to learning the kernel results in a linear combination of positive semidefinite kernels. However, an arbitrary linear combination of positive kernels is not necessarily positive semidefinite [Mary, 2003]. For example, a class formed by scaled versions of a single positive semidefinite kernel will have negative kernels as well, when the scalar coefficient is negative. While the elements of the associated vector space of kernels can always be defined as the difference between two positive kernels, what is the functional space associated with such a kernel?

The purpose of this chapter is to point out that there is a much larger class of kernel functions available, which do not necessarily correspond to a RKHS but which nonetheless can be used for machine learning. Such kernels are known as *indefinite kernels*, as the scalar product matrix may contain a mix of positive and negative eigenvalues. Apart from the above motivation, there are several other independent reasons for studying indefinite kernels:

- Testing Mercer's condition for a given kernel can be a challenging task which may well lie beyond the abilities of a practitioner.
- Sometimes functions which can be proven *not* to satisfy Mercer's condition may be of interest. One such instance is the hyperbolic tangent kernel $k(x, x') = \tanh(\langle x, x' \rangle - 1)$ of Neural Networks [Haykin, 1999], which is indefinite for any range of parameters or dimensions [Smola et al., 2000].
- There have been promising empirical reports on the use of indefinite kernels [Lin and Lin, 2003].
- In H^∞ control applications and discriminant analysis, the cost function can be formulated as the difference between two quadratic norms [Haasdonk, 2003, Hassibi et al., 1999], corresponding to an indefinite inner product.
- The solution of partial differential equations arising from the Navier-Stokes equations for fluid flow results in an indefinite problem.
- RKKS theory (concerning function spaces arising from indefinite kernels) has become a rather active area in interpolation and approximation theory [Dritschel and Rovnyak, 1996, Alpay et al., 1997, Rovnyak, 2002].

We will discuss the above issues using topological spaces similar to Hilbert spaces except for the fact that the inner product is no longer necessarily positive.

4.2 Reproducing Kernel Krein Spaces

Kreĭn spaces are indefinite inner product spaces endowed with a Hilbertian topology, yet their inner product is no longer positive. Before we delve into definitions and state basic properties of Kreĭn spaces, we give an example:

Example 16 (4 dimensional space-time) *Indefinite spaces were first used by Minkowski for the solution of problems in special relativity. There the inner product in space-time (x, y, z, t) is given by*

$$\langle (x, y, z, t), (x', y', z', t') \rangle = xx' + yy' + zz' - tt'.$$

Observe that it is not positive. The vector $v = (1, 1, 1, \sqrt{3})$ belongs to the cone of so-called neutral vectors which satisfy $\langle v, v\rangle = 0$ (in coordinates $x^2 + y^2 + z^2 - t^2 = 0$). In special relativity this cone is also called the "light cone," as it corresponds to the propagation of light from a point event.

4.2.1 Kreĭn spaces

The above example shows that there are several differences between Kreĭn spaces and Hilbert Spaces. We now define Kreĭn spaces formally. More detailed expositions can be found in Bognár [1974] and Azizov and Iokhvidov [1989]. The key difference is the fact that the inner products are indefinite.

Definition 16 (Inner product) *Let $\mathcal{K}$ be a vector space on the scalar field.*[1] *An inner product $\langle ., .\rangle_{\mathcal{K}}$ on $\mathcal{K}$ is a bilinear form where for all $f, g, h \in \mathcal{K}$, $\alpha \in \mathbb{R}$:*

- $\langle f, g\rangle_{\mathcal{K}} = \langle g, f\rangle_{\mathcal{K}}$
- $\langle \alpha f + g, h\rangle_{\mathcal{K}} = \alpha\langle f, h\rangle_{\mathcal{K}} + \langle g, h\rangle_{\mathcal{K}}$
- $\langle f, g\rangle_{\mathcal{K}} = 0$ *for all* $g \in \mathcal{K}$ *implies* $\Rightarrow f = 0$

An inner product is said to be *positive* if for all $f \in \mathcal{K}$ we have $\langle f, f\rangle_{\mathcal{K}} \geq 0$. It is *negative* if for all $f \in \mathcal{K}$ $\langle f, f\rangle_{\mathcal{K}} \leq 0$. Otherwise it is called *indefinite*.

A vector space $\mathcal{K}$ embedded with the inner product $\langle ., .\rangle_{\mathcal{K}}$ is called an *inner product space*. Two vectors f, g of an inner product space are said to be *orthogonal* if $\langle f, g\rangle_{\mathcal{K}} = 0$. Given an inner product, we can define the associated space.

Definition 17 (Kreĭn space) *An inner product space $(\mathcal{K}, \langle ., .\rangle_{\mathcal{K}})$ is a Kreĭn space if there exist two Hilbert spaces $\mathcal{H}_+, \mathcal{H}_-$ spanning $\mathcal{K}$ such that*

- *All $f \in \mathcal{K}$ can be decomposed into $f = f_+ + f_-$, where $f_+ \in \mathcal{H}_+$ and $f_- \in \mathcal{H}_-$.*
- $\forall f, g \in \mathcal{K}, \langle f, g\rangle_{\mathcal{K}} = \langle f_+, g_+\rangle_{\mathcal{H}_+} - \langle f_-, g_-\rangle_{\mathcal{H}_-}$

This suggests that there is an associated Hilbert space, where the difference in scalar products is replaced by a sum:

Definition 18 (Associated Hilbert Space) *Let $\mathcal{K}$ be a Kreĭn space with decomposition into Hilbert spaces $\mathcal{H}_+$ and $\mathcal{H}_-$. Then we denote by $\overline{\mathcal{K}}$ the associated Hilbert space defined by*

$$\overline{\mathcal{K}} = \mathcal{H}_+ \oplus \mathcal{H}_- \text{ hence } \langle f, g\rangle_{\overline{\mathcal{K}}} = \langle f_+, g_+\rangle_{\mathcal{H}_+} + \langle f_-, g_-\rangle_{\mathcal{H}_-}$$

[1] Like Hilbert spaces, Kreĭn spaces can be defined on $\mathbb{R}$ or $\mathbb{C}$. We use $\mathbb{R}$ in this thesis.

Likewise we can introduce the symbol $\ominus$ to indicate that

$$\mathcal{K} = \mathcal{H}_+ \ominus \mathcal{H}_- \text{ hence } \langle f, g\rangle_{\mathcal{K}} = \langle f_+, g_+\rangle_{\mathcal{H}_+} - \langle f_-, g_-\rangle_{\mathcal{H}_-}.$$

$\overline{\mathcal{K}}$ is the smallest Hilbert space majorizing the Kreĭn space $\mathcal{K}$ and one defines the strong topology on $\mathcal{K}$ as the Hilbertian topology of $\overline{\mathcal{K}}$. The topology does not depend on the decomposition chosen. In fact the majorization means that $|\langle f, f\rangle_{\mathcal{K}}| \leqslant \|f\|^2_{\overline{\mathcal{K}}}$ for all $f \in \mathcal{K}$. The decomposition $\mathcal{K} = \mathcal{H}_+ \ominus \mathcal{H}_-$ is called the fundamental decomposition of the Kreĭn space $\mathcal{K}$, and associated with it is the fundamental symmetry $\mathbf{J}$ defined for $f_+ \in \mathcal{H}_+$ and $f_- \in \mathcal{H}_-$, as $\mathbf{J}(f_+ + f_-) = f_+ - f_-$. In essence the fundamental symmetry can be thought of as the identity operator in Kreĭn spaces.

Definition 19 (Adjoint Operator) *Let $T : \mathcal{K}_1 \to \mathcal{K}_2$ be a bounded linear operator between Kreĭn spaces $\mathcal{K}_1$ and $\mathcal{K}_2$. Then the* adjoint operator T^* *of T is defined to be the unique operator such that [Bognár, 1974, Section VI.2]*

$$\langle Tf, g\rangle_{\mathcal{K}_2} = \langle f, T^* g\rangle_{\mathcal{K}_1} \text{ where } f \in \mathcal{K}_1, g \in \mathcal{K}_2.$$

Riesz representation theorem holds in Kreĭn spaces [Mary, 2003, Theorem 2.9] and therefore (like Hilbert spaces) Kreĭn spaces are self dual.

The space $\mathcal{K}$ is said to be *Pontryagin* if it admits a decomposition with finite dimensional $\mathcal{H}_-$, and *Minkowski* if $\mathcal{K}$ itself is finite dimensional. Note that every finite dimensional inner product space is decomposable, therefore every finite dimensional non-degenerate inner product space is a Kreĭn space [Bognár, 1974, Section I.11]. We will see how Pontryagin spaces arise naturally when dealing with conditionally positive definite kernels (see Section 4.2.4).

For estimation we need to introduce Kreĭn spaces on functions. Let $\mathcal{X}$ be the learning domain, and $\mathbb{R}^{\mathcal{X}}$ the set of functions from $\mathcal{X}$ to $\mathbb{R}$. The evaluation functional tells us the value of a function at a certain point, and we shall see that the RKKS is a subset of $\mathbb{R}^{\mathcal{X}}$ where this functional is continuous.

Definition 20 (Evaluation functional)

$$T_x : \mathcal{K} \to \mathbb{R} \text{ where } f \mapsto T_x f = f(x).$$

Definition 21 (RKKS) *A Kreĭn space $(\mathcal{K}, \langle ., .\rangle_{\mathcal{K}})$ is a Reproducing Kernel Kreĭn Space [Alpay, 2001, Chapter 7] if $\mathcal{K} \subset \mathbb{R}^{\mathcal{X}}$ and the evaluation functional is continuous on $\mathcal{K}$ endowed with its strong topology (that is, via $\overline{\mathcal{K}}$).*

4.2.2 From Kreĭn spaces to Kernels

We prove an analog to the Moore-Aronszajn theorem [Wahba, 1990], which tells us that for every kernel there is an associated Kreĭn space, and for every RKKS, there is a unique kernel.

Proposition 22 *Let $\mathcal{K}$ be an RKKS with $\mathcal{K} = \mathcal{H}_+ \ominus \mathcal{H}_-$. Then*

1. *$\mathcal{H}_+$ and $\mathcal{H}_-$ are RKHS (with kernels k_+ and k_-),*
2. *There is a unique symmetric $k(x, x')$ with $k(x, \cdot) \in \mathcal{K}$ such that for all $f \in \mathcal{K}$, $\langle f, k(x, \cdot)\rangle_{\mathcal{K}} = f(x)$,*
3. *$k = k_+ - k_-$.*

Proof Since $\mathcal{K}$ is a RKKS, the evaluation functional is continuous with respect to the strong topology. Hence the associated Hilbert Space $\overline{\mathcal{K}}$ is an RKHS. It follows that $\mathcal{H}_+$ and $\mathcal{H}_-$, as Hilbertian subspaces of an RKHS, are RKHS themselves with kernels k_+ and k_- respectively. Let $f = f_+ + f_-$. Then $T_x(f)$ is given by

$$\begin{aligned} T_x(f) &= T_x(f_+) + T_x(f_-) \\ &= \langle f_+, k_+(x, \cdot)\rangle_{\mathcal{H}_+} - \langle f_-, -k_-(x, \cdot)\rangle_{\mathcal{H}_-} \\ &= \langle f, k_+(x, \cdot) - k_-(x, \cdot)\rangle_{\mathcal{K}}. \end{aligned}$$

In both lines we exploited the orthogonality of $\mathcal{H}_+$ with $\mathcal{H}_-$. Since k_+ and k_- are symmetric, $k := k_+ - k_-$ is symmetric. Since the inner product $\langle \cdot, \cdot \rangle_{\mathcal{K}}$ is non-degenerate, k is unique. ■

4.2.3 From Kernels to Kreĭn spaces

Let k be a symmetric real valued function on $\mathcal{X}^2$.

Proposition 23 *The following are equivalent [Mary, 2003, Theorem 2.28]:*

- *There exists (at least) one RKKS with kernel k.*
- *k admits a positive decomposition, that is there exists two positive kernels k_+ and k_- such that $k = k_+ - k_-$.*
- *k is dominated by some positive kernel p (that is, $p - k$ is a positive kernel).*

There is *no* bijection but a surjection between the set of RKKS and the set of generalized kernels defined in the vector space generated out of the cone of positive kernels.

4.2.4 Examples and Spectral Properties

We collect several examples of indefinite kernels in Table 4.1 and plot a 2 dimensional example as well as 20 of the eigenvalues with the largest absolute value. We investigate the spectrum of radial kernels using the Hankel transform.

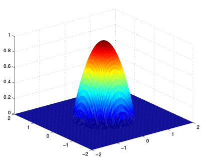
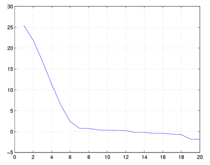
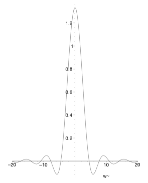
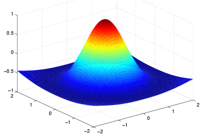
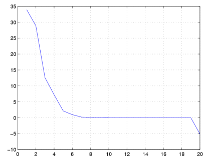
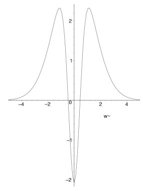
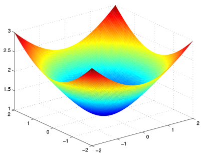
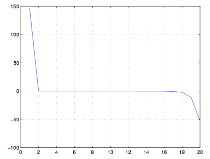
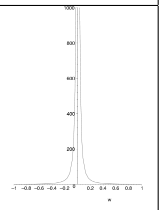
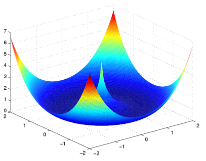
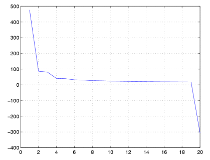
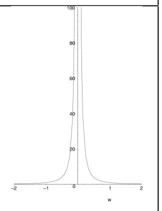
Kernel	2D kernel	20 Eigenvalues	Spectra
Epanechnikov kernel $\left(1-\frac{\|s-t\|^2}{\sigma}\right)^p$, for $\frac{\|s-t\|^2}{\sigma} \leqslant 1$			
Gaussian Combination $\exp\left(\frac{-\|s-t\|^2}{\sigma_1}\right)+\exp\left(\frac{-\|s-t\|^2}{\sigma_2}\right)$ $-\exp\left(\frac{-\|s-t\|^2}{\sigma_3}\right)$			
Multiquadric kernel $\sqrt{\frac{\|s-t\|^2}{\sigma}+c^2}$			
Thin plate spline $\frac{\|s-t\|}{\sigma}^{2p}\ln\left(\frac{\|s-t\|^2}{\sigma}\right)$			

Table 4.1: Examples of indefinite kernels. Column 2 shows the 2D surface of the kernel with respect to the origin, column 3 shows plots of the 20 eigenvalues with largest magnitude of uniformly spaced data from the interval $[-2, 2]$, column 4 shows plots of the Fourier spectra.

Isometries commuting with symmetry groups, such as the Fourier transform or decompositions according to spherical harmonics are ideally suited to analyzing the spectral properties of kernels. The Fourier transform allows one to find the eigenvalue decomposition of kernels of the form $k(x, x') = h(x - x')$ by computing the Fourier transform of h. For $x \in \mathbb{R}^n$ we have

$$F[f](\|\omega\|) = \|\omega\|^{-\nu} H_\nu[r^\nu h(r)](\|\omega\|),$$

where $\nu = \frac{1}{2}n - 1$ and H_ν is the Hankel transform of order ν. Bochner's Theorem (for example Reed and Simon [1980, Theorem IX.9], and as used in Smola et al. [1998]) tells us that a kernel is positive semidefinite if and only if its Fourier spectrum is positive. Table 4.1 depicts the spectra of these kernels. Negative values in the Hankel transform correspond to $\mathcal{H}_-$, positive ones to $\mathcal{H}_+$. Likewise the decomposition of $k(x, x') = h(\langle x, x' \rangle)$ in terms of associated Legendre polynomials allows one to identify the positive and negative parts of the Kreĭn space, as the Legendre polynomials commute with the rotation group [Smola et al., 2000].

One common class of translation invariant kernels which are not positive definite are so-called conditionally positive definite (cpd) kernels. A cpd kernel of order p leads to a positive semidefinite matrix in a subspace of coefficients orthogonal to polynomials of order up to $p-1$. Moreover, in the subspace of $(p-1)$ degree polynomials, the inner product is typically negative definite. This means that there is a space of polynomials of degree up to order $p-1$ (which constitutes an up to $\binom{n+p-2}{p-1}$-dimensional subspace) with negative inner product. In other words, we are dealing with a Pontryagin space.

The standard procedure to use such kernels is to project out the negative component, replace the latter by a suitably smoothed estimate in the polynomial subspace and treat the remaining subspace as any RKHS [Wahba, 1990]. Using Kreĭn spaces we can use these kernels directly, without the need to deal with the polynomial parts separately.

4.3 Machine Learning in RKKS

In this section we show how to express machine learning problems with indefinite kernels. In order to perform machine learning, we would like to optimize over a class of functions, and also to prove that the solution exists and is unique. Instead of minimizing over a class of functionals as in a RKHS, we look for the stationary point. It turns out that this point can be found using projections onto the subspace spanned by the kernel functions evaluated on the data. Although we can derive the general framework for Tikhonov regularization for Kreĭn space, this approach as several practical difficulties. We will present an alternative form of regularization in Chapter 5.

Researchers have been aware of the limitation of positive semidefinite kernels, and several attempts have been made to extend the class of kernels. Schölkopf [2001] proposed a framework for machine learning with conditionally positive definite kernels, which as mentioned above, define a Pontryagin space. This framework was further investigated in Pekalska et al. [2001]. The motivations for these papers were to generalize kernels to measure dissimilarities. Graepel et al. [1999] performed learning where the measure between objects was indefinite. However, their approach assumed the knowledge of the positive and negative parts of the spectrum.

4.3.1 Projection and Stabilization

Given some data $X_{\text{train}} = \{x_1, \dots, x_m\}$, and labels $Y_{\text{train}} = \{y_1, \dots, y_m\}$, we consider the interpolation problem of finding the function $f(x_i) = y_i$. Let $T : \mathcal{K} \to \mathbb{R}^m$ be the evaluation functional (Definition 20) where $f \mapsto Tf = [f(x_1), \dots, f(x_m)]^\top$, that is T maps the function f from the RKKS $\mathcal{K}$ to the vector consisting of the evaluation of this function on the input data $x_i \in \mathcal{X}$. Note that $\mathbb{R}^m$ has the standard Hilbertian topology defined on it. Hence the interpolation problem is finding a function $f \in \mathcal{K}$ such that for $y \in \mathbb{R}^m$,

$$Tf = y. \tag{4.1}$$

In other words, for the given training labels $y = Y_{\text{train}}$, we want a function f such that the evaluation fits the labels. There may be numerous solutions to this problem. In Hilbert space, the solution is to choose the function with minimal norm. This is called the minimal norm interpolation problem. Details of the minimal norm interpolation problem and its relationship to the pseudo-inverse are described in Albert [1972] and Groetsch [1977]. We derive the details of interpolation in Kreĭn spaces. Observe that the difficulty lies in the fact that we do not have a norm in Kreĭn space.

Define $T^* : \mathbb{R}^m \to \mathcal{K}$, the adjoint operator of T such that $\langle Tf, \alpha \rangle_{\mathbb{R}^m} = \langle f, T^*\alpha \rangle_{\mathcal{K}}$, where $\alpha \in \mathbb{R}^m$. From the left hand side,

$$\begin{aligned}
\langle Tf, \alpha \rangle_{\mathbb{R}^m} &= \sum_{i=1}^m \alpha_i f(x_i) && \text{(by the definition of } T\text{)} \\
&= \sum_{i=1}^m \alpha_i \langle f(\cdot), k(\cdot, x_i) \rangle_{\mathcal{K}} && \text{(}\mathcal{K}\text{ is a reproducing kernel Kreĭn space)} \\
&= \langle f(\cdot), \sum_{i=1}^m \alpha_i k(\cdot, x_i) \rangle_{\mathcal{K}} && \text{(by linearity of the inner product)}
\end{aligned}$$

Therefore the adjoint maps $\alpha \in \mathbb{R}^m$ to $T^*\alpha = \sum_{i=1}^m \alpha_i k(x_i, \cdot)$, where $k(\cdot, \cdot)$ is the kernel associated with $\mathcal{K}$. The operator TT^* has a special meaning

$$\begin{aligned}
TT^*\alpha &= T \sum_{i=1}^m \alpha_i k(x_i, \cdot) \\
&= \left[\sum_{i=1}^m \alpha_i k(x_i, x_1), \dots, \sum_{i=1}^m \alpha_i k(x_i, x_m) \right]^\top \\
&= K\alpha
\end{aligned} \tag{4.2}$$

where K is the Gram matrix defined by $K_{ij} = k(x_i, x_j)$. For simplicity, we assume that K is regular. Hence, the operator TT^* is actually the finite dimensional gram matrix K.

Using the notation defined above, we investigate the variational approach. The variational formulation expresses the primal problem as an equality for all dual variables. Since the dual space of $\mathbb{R}^m$ is itself, given T and y, we want to find f such that

$$\langle v, Tf\rangle_{\mathbb{R}^m} = \langle v, y\rangle_{\mathbb{R}^m} \text{ for all } v \in \mathbb{R}^m. \tag{4.3}$$

The solution of Equation (4.3) is called a weak solution of Equation (4.1). The following proposition shows that the solution to the interpolation problem is the orthogonal projection.

Proposition 24 *Define the solution set $\mathcal{F} = \{f \in \mathcal{K}$ such that $Tf = y\}$, that is for all $f \in \mathcal{F}$, $Tf = y$. Define the finite dimensional solution set $\mathcal{A} = \{\alpha \in \mathbb{R}^m$ such that $K\alpha = y\}$. Then $\mathcal{F}$ is the set $\{T^*\alpha + u$ where $\alpha \in \mathcal{A}$ and $u \in \mathcal{K}$ such that $Tu = 0\}$.*

Proof From Equation (4.3), for all $v \in \mathbb{R}^m$, $g \in \mathcal{F}$ and $f \in \mathcal{K}$ such that $f \neq g$,

$$\begin{aligned}
\langle v, Tf - y\rangle_{\mathbb{R}^m} &= 0\\
\langle v, Tf - Tg\rangle_{\mathbb{R}^m} &= 0\\
\langle v, T(f - g)\rangle_{\mathbb{R}^m} &= 0\\
\langle T^*v, f - g\rangle_{\mathcal{K}} &= 0.
\end{aligned}$$

Observe that the above equation is true for all $v \in \mathbb{R}^m$ and $g \in \mathcal{F}$, hence T^*v is orthogonal to $f - g$. Therefore f is the orthogonal projection, in $\mathcal{K}$, of g onto $\mathcal{R}(T^*)$. This means that there exists an $\alpha \in \mathbb{R}^m$ such that $f = T^*\alpha + u$ where $Tu = 0$. Hence solving $Tf = y$ is equivalent to solving $TT^*\alpha = y$, and by Equation (4.2) we get the result. ■

In a Hilbert space, then the orthogonal projection is given by [Brezinski, 1997],

$$\begin{aligned}
\min_{\alpha\in\mathbb{R}^m} \frac{1}{2}\|T^*\alpha - g\|^2 &= \min_{\alpha\in\mathbb{R}^m} \frac{1}{2}\alpha^\top TT^*\alpha - \alpha^\top Tg\\
&= \min_{\alpha\in\mathbb{R}^m} \frac{1}{2}\alpha^\top K\alpha - \alpha^\top y.
\end{aligned} \tag{4.4}$$

In the case of Kreĭn spaces, Equation (4.4) does not have a solution, since we have a possibly negative norm. The following calculation demonstrates this phenomenon. Denote the solution of Equation (4.1) by f^*, where $f^* \in \mathcal{K}$ and $y \in \mathbb{R}^m$, and the corresponding coefficients by α^*. Define a quadratic functional

$$J(\alpha) = \frac{1}{2}\langle \alpha, K\alpha\rangle - \langle y, \alpha\rangle. \tag{4.5}$$

By considering the behavior of the optimal value of $\alpha = \alpha^*$, we can see that we cannot minimize Equation (4.5). Let $e = \alpha - \alpha^*$, be the error of our estimate from its optimal value. Then, using (4.5),

$$\begin{aligned} J(\alpha) &= \tfrac{1}{2}\langle \alpha^* + e, K(\alpha^* + e)\rangle - \langle y, \alpha^* + e\rangle \\ &= \tfrac{1}{2}\langle \alpha^*, K\alpha^*\rangle - \langle y, \alpha^*\rangle + \tfrac{1}{2}\langle e, Ke\rangle - \langle e, K\alpha^*\rangle - \langle y, e\rangle \quad \text{(symmetry)} \\ &= J(\alpha^*) + \tfrac{1}{2}\langle e, Ke\rangle. \quad (K\alpha^* = y) \end{aligned}$$

Since K is indefinite, the second term on the right hand side can be negative, and therefore, unlike the case when K is positive, $J(\alpha^*)$ is not the minimizer of (4.5). However, we can still find the stationary point. Observe that if K is positive, we obtain the orthogonal projection as $J(\alpha)$ is equal to Equation (4.4). Therefore, for indefinite K, we solve for the stationary point of $J(\alpha)$ as defined by Equation (4.5).

Proposition 25 *The solution to Equation (4.1), is given by $T^*\alpha$, where $\alpha = \alpha^*$ is the stationary point of $J(\cdot)$ (Equation 4.5). At the stationary point, for invertible K, $J(\alpha^*) = -\frac{1}{2}\langle \alpha^*, K\alpha^*\rangle = \frac{1}{2}\langle y, K^{-1}y\rangle$.*

Proof Since f^* is a solution of Equation (4.1), $TT^*\alpha^* = K\alpha^* = y$. Substituting this into the gradient, we get $\nabla J(\alpha^*) = K\alpha^* - y = 0$, which shows that $J(\cdot)$ is stationary at α^*. From Equation (4.5),

$$J(\alpha^*) = \frac{1}{2}\langle \alpha^*, K\alpha^*\rangle - \langle y, \alpha^*\rangle = -\frac{1}{2}\langle \alpha^*, K\alpha^*\rangle,$$

since $\langle y, \alpha^*\rangle = \langle \alpha^*, K\alpha^*\rangle$. Furthermore, for regular K, $\alpha^* = K^{-1}y$, which gives the second equality. ■

The results above show that the optimum point Equation (4.1) can be found by finding the stationary point of Equation (4.5). In fact, we can make a much more general statement when performing stabilization.

4.3.2 Application to general spline smoothing

We consider the general spline smoothing problem as presented in Wahba [1990], except we are considering Kreĭn spaces. The general spline smoothing is defined as the function stabilizing (that is finding the stationary point) the following criterion:

$$J(f) = \frac{1}{m}\sum_{i=1}^{m}\big(y_i - f(x_i)\big)^2 + \lambda\langle f, f\rangle_{\mathcal{K}}. \tag{4.6}$$

The solution can be found using,

$$\begin{aligned} J(f) \quad &= \tfrac{1}{2}\langle Tf - y, Tf - y\rangle_{\mathbb{R}^m} + \tfrac{\lambda}{2}\langle f, f\rangle_{\mathcal{K}} \\ &= \tfrac{1}{2} f^\top T^\top T f - y^\top T f + \tfrac{1}{2} y^\top y + \tfrac{\lambda}{2}\langle f, f\rangle_{\mathcal{K}}. \end{aligned}$$

The gradient is given by

$$\nabla J(f) = (T^\top T + \lambda \mathbf{J}) f - T^\top y,$$

where $\mathbf{J}$ is the fundamental symmetry, with possibly ±1 along the diagonal, and hence the stationary point occurs at

$$(T^\top T + \lambda \mathbf{J}) f = T^\top y.$$

Note that the effect of λ no longer necessarily improves the condition number of the linear problem. In fact, if λ is equal to an eigenvalue of $T^\top T$, the system is singular.

We will investigate the smoothing problem in further detail in Section 5.3.2 and Section 6.2.

4.4 Generalization Bounds via Rademacher Average

An important issue regarding learning algorithms are their ability to generalize (to give relevant predictions). This property is obtained when the learning process considered shows an uniform convergence behaviour. In Mendelson [2003] such a result is demonstrated in the case of RKHS through the control of the Rademacher average of the class of function considered. Here we present an adaptation of this proof in the case of Kreĭn spaces. We begin with setting the functional framework for the result.

Let k be a kernel defined on a set $\mathcal{X}$ and choose a decomposition $k = k_+ - k_-$ where k_+ and k_- are both positive kernels. This given decomposition of the kernel can be associated with the RKHS $\overline{\mathcal{K}}$ defined by its positive kernel $\overline{k} = k_+ + k_-$ whose Hilbertian topology defines the strong topology of $\mathcal{K}$. We will then consider the set $\mathcal{B}_\mathcal{K}$ defined as follows:

$$\mathcal{B}_\mathcal{K} = \left\{ f \in \mathcal{K} \middle| \; \|f_+\|^2 + \|f_-\|^2 = \|f\|^2 \leq 1 \right\}$$

Note that in a Kreĭn space the norm of a function is the associated Hilbertian norm and usually $\|f\|^2 \neq \langle f, f\rangle_\mathcal{K}$ but always $\langle f, f\rangle_\mathcal{K} \leq \|f\|^2$.

The Rademacher average of a class of functions $\mathcal{F}$ with respect to a measure μ is defined as follows. Let $x_1, \dots, x_m \in \mathcal{X}$ be i.i.d random variables sampled according to μ. Let ε_i for $i = 1, \dots, m$ be Rademacher random variables, that is variables taking values $\{-1,+1\}$ with equal probability.

Definition 26 (Rademacher Average) *The* Rademacher average, *$R_m(\mathcal{F})$ of a set of functions $\mathcal{F}$ (w.r.t. μ) is defined as*

$$R_m(\mathcal{F}) = \mathbb{E}_\mu \mathbb{E}_\varepsilon \frac{1}{\sqrt{m}} \sup_{f\in\mathcal{F}} \Big| \sum_{i=1}^{m} \varepsilon_i f(x_i) \Big|.$$

Using the Rademacher average as an estimate of the "size" of a function class, we can obtain generalization error bounds which are also called uniform convergence or sample complexity bounds [Mendelson, 2003, Corollary 3], that is for any $\varepsilon > 0$ and $\delta > 0$, there is an absolute constant C such that if

$$m \geqslant \frac{C}{\varepsilon^2} \max\{R_m^2(J(\mathcal{B}_\mathcal{K})), \log\frac{1}{\delta}\}, \tag{4.7}$$

then

$$\mathbb{P}\mathbf{r}\Big(\sup_{f\in\mathcal{B}_\mathcal{K}} \Big| \frac{1}{m}\sum_{i=1}^{m} J(f(X_i)) - \mathbb{E}J(f) \Big| \geq \varepsilon \Big) \leq \delta \tag{4.8}$$

where $J(f(x))$ denotes the quadratic loss defined as in Mendelson [2003]. To get the expected result we have to show that the Rademacher average is bounded by a constant independent of the sample size m. To control the Rademacher average, we first give a lemma regarding the topology of Kreĭn spaces putting emphasis on both difference and close relationship with the Hilbertian case.

Lemma 27 *For all $g \in \mathcal{K}$:*

$$\sup_{f\in\mathcal{B}_\mathcal{K}} \langle f(.), g(.)\rangle_\mathcal{K} = \|g\|$$

Proof It is trivial if $g = 0$. $\forall g \in \mathcal{K}$, $g \neq 0$, let $h = g/\|g\|$. By construction $\|h\| = 1$.

$$\begin{aligned}
\sup_{f\in\mathcal{B}_\mathcal{K}} \langle f(.), g(.)\rangle_\mathcal{K} &= \|g\| \sup_{f\in\mathcal{B}_\mathcal{K}} \langle f(.), h(.)\rangle_\mathcal{K} \\
&= \|g\| \sup_{f\in\mathcal{B}_\mathcal{K}} \big(\langle f_+, h_+\rangle_{\mathcal{K}_+} - \langle f_-, h_-\rangle_{\mathcal{K}_-}\big) \\
&= \|g\| \big(\langle h_+, h_+\rangle_{\mathcal{K}_+} + \langle h_-, h_-\rangle_{\mathcal{K}_-}\big) \\
&= \|g\|
\end{aligned}$$

■

In the unit ball of a RKKS, the Rademacher average with respect to the probability measure μ behaves the same way as the one of its associated RKHS.

Proposition 28 (Rademacher Average of an Indefinite Kernel) *Let $\overline{K}$ be the Gram matrix of kernel $\overline{k}$ at points $x_1, \ldots, x_m$. If according to the measure μ on $\mathcal{X}$ $x \longmapsto \overline{k}(x,x) \in L^1(\mathcal{X}, \mu)$, then*

$$R_m(\mathcal{B}_{\mathcal{K}}) \leq M^{\frac{1}{2}}$$

with

$$M = \frac{1}{m}\,\mathbb{E}_{\mu}\big(tr\,(\overline{K})\big) = \int_{\mathcal{X}} \overline{k}(x,x)d\mu(x)$$

The proof works just as in the Hilbertian case [Mendelson, 2003, Theorem 16] with the application of Lemma 27, and is shown in Appendix B. As a second slight difference we choose to express the bound as a function of the $L^1(\mathcal{X}, \mu)$ norm of the kernel instead of going through its spectral representation. It is simpler since for instance, for the unnormalized Gaussian kernel $k(x,y) = \exp(-(x-y)^2)$ on $\mathcal{X} = \mathbb{R}$ we have $M = 1$ regardless the measure μ considered.

Hence from Proposition 28, we can bound the generalization error by using Equations (4.7) and (4.8).

4.5 Spectrum of the Evaluation Operator

If A is a compact linear self-adjoint operator on a Kreĭn space, then A can be diagonalized [Azizov and Iokhvidov, 1989, Chapter 4]. Furthermore, operators of the form $A = T^*T$ can be fundamentally decomposed into its positive and negative parts [Bognár, 1974, Chapter VII]. In this section, we investigate the special case when we have $A = T^*T$ where T is an evaluation operator from a Kreĭn space to a Hilbert space. These results will be useful for the analysis of regularization methods (Section 5.3) and reconstruction error bounds for principal component analysis (Section 6.1). In particular we will analyze various regularization methods in terms of the spectrum of the kernel and use Proposition 30 in the proof of Theorem 38.

Recall that $TT^* = K$. From the spectral decomposition theorem (see Golub and van Loan [1996, Theorem 8.1.1] for the matrix version and Reed and Simon [1980, Chapter 7] for the general Hilbert space version), TT^* has the eigenvalue decomposition $TT^* = U\Lambda U^\top$, where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$, is the diagonal matrix of the eigenvalues λ of TT^*, and the column vectors of U are the corresponding eigenvectors $u_1, \ldots, u_m$. Therefore, we can write $TT^* u_i = \lambda_i u_i$, for $i = 1, \ldots, m$.

For a Kreĭn space $\mathcal{K}$, let $\mathcal{L}(\mathcal{K}) = \{T : \mathcal{K} \to \mathcal{K}\}$ be the class of continuous linear operators. If $A \in \mathcal{L}(\mathcal{K})$ is a compact self-adjoint operator on a Kreĭn space, there exists an orthonormal basis $\{e_i\}$ of $\mathcal{K}$ of eigenfunctions of A. Let $\mathrm{spec}(A)$ be the spectrum of such an operator, sorted in non-increasing order of magnitude $|\lambda_1(A)| \geqslant |\lambda_2(A)| \geqslant \ldots$. An operator A is called trace class if $\sum_{i \geqslant 1} \langle Ae_i, e_i \rangle_{\mathcal{K}}$ is a convergent series. We denote the trace of the operator A by $\mathrm{tr}A = \sum_{i\geqslant 1}\langle Ae_i, e_i\rangle_{\mathcal{K}}\mathrm{sign}\langle e_i, e_i\rangle_{\mathcal{K}}$ [Azizov and Iokhvidov, 1989, pg. 223]. We would like to relate the spectrum of the kernel operator with the spectrum of the covariance operator, as done in the finite dimensional context by Shawe-Taylor et al. [2002] and in the Hilbert space context by Zwald et al. [2004].

Note that unlike the Hilbertian case, the spectrum of a self-adjoint operator in Kreĭn space need not be real, as the following example shows.

Example 17 *[Bognár, 1974, pg. 35] Let $\{e, f\}$ be a basis of the indefinite inner product space $\mathcal{X}$, where*

$$\langle e, e\rangle = \langle f, f\rangle = 0, \langle e, f\rangle = 1.$$

A linear operator specified by the relations

$$Ae = ie, Af = -if$$

is symmetric.

However, since the operator T has a range which is a subset of a Hilbert space, we can show that the operator T^*T has real eigenvalues. Moreover, the spectrums of TT^* and T^*T are the same.

Proposition 29 *Let v_i be the set of orthogonal vectors in $\mathcal{K}$, defined as*

$$v_i := \frac{1}{\sqrt{|\lambda_i|}}T^*u_i,$$

then

$$T^*Tv_i = \lambda_i v_i \text{ and } \mathrm{spec}(TT^*) \cup \{0\} = \mathrm{spec}(T^*T) \cup \{0\}.$$

Proof For v_i, v_j corresponding to eigenvectors $u_i, u_j \in \mathbb{R}^m$, for $i \neq j$, $\langle v_i, v_j\rangle = 0$, since $\langle u_i, u_j\rangle = 0$. However, unlike the positive semidefinite case, since λ_i is not necessarily non-negative, $\langle v_i, v_i\rangle = \pm 1$. From the definition of v_i, we obtain $T^*u_i = \sqrt{|\lambda_i|}v_i$, and $Tv_i = \frac{\lambda_i}{\sqrt{|\lambda_i|}}u_i$. Hence $T^*Tv_i = \lambda_i v_i$. Observe that the spectrum of T^*T is the same as the spectrum of TT^*, with the possibility of zero being part of the spectrum of either one. ■

When we are dealing with a random variable X that is independent and identically distributed according to P, there corresponds a space of functions determined by $k(X, \cdot)$ where k is the reproducing kernel. The covariance operator of these functions and the kernel operator

$$(Kf)(x) = \int f(y)k(x, y)dP(y),$$

have the same spectrum.

Proposition 30 *Let $T : \mathcal{K} \to L_2$ be the evaluation operator, and T^* its adjoint. Let $k(\cdot,\cdot)$ be a kernel such that $\mathbb{E}(k(X,X)) < \infty$. Then, $C = T^*T$ is the covariance operator, and $K = TT^*$ is the integral operator associated with the kernel $k(\cdot,\cdot)$. Furthermore,* $\mathrm{spec}(C) = \mathrm{spec}(K)$ *and K is a self-adjoint trace class operator and* $\mathrm{tr}K = \sum_{i\geqslant 1}\lambda_i(K)$.

Proof Following the proof of Zwald et al. [2004, Theorem 1], we get $T^*(f) = \mathbb{E}[f(X)k(X,\cdot)]$ for a random variable X. By definition of expectation, for all $f, f' \in \mathcal{K}$,

$$\begin{aligned}\langle f, T^*Tf'\rangle &= \langle f, \mathbb{E}[k(X,\cdot)\langle k(X,\cdot), f'\rangle]\rangle\\ &= \mathbb{E}[\langle k(X,\cdot), f\rangle\langle k(X,\cdot), f'\rangle].\end{aligned}$$

Hence, by uniqueness of the covariance operator, we get $C = T^*T$. Similarly,

$$\begin{aligned}(TT^*f)(x) &= \langle T^*f, k(x,\cdot)\rangle\\ &= \mathbb{E}[\langle f(X)k(X,\cdot), k(X,\cdot)\rangle]\\ &= \textstyle\int f(y)k(x,y)dP(y).\end{aligned}$$

Since T is a compact operator, by the spectral decomposition $\mathrm{spec}(C) = \mathrm{spec}(K)$ and $\mathrm{tr}C = \mathrm{tr}K = \sum_{i\geqslant 1}\lambda_i(K)$. ■

Note that although the covariance operator C has possibly negative eigenvalues, for any $f \in \mathcal{K}$, $\langle f, Cf\rangle_{\mathcal{K}} \geqslant 0$. Corresponding to the eigenvalues of TT^*, we can define the notion of a singular value, and hence the singular value decomposition. Denote the tensor product between vectors by $\otimes$, that is for $u, x \in \mathbb{R}^m$ and $v, f \in \mathcal{K}$, $(v \otimes u)x = \langle u, x\rangle_{\mathbb{R}^m} v$, and $(u \otimes v)f = \langle v, f\rangle_{\mathcal{K}} u$.

Proposition 31 *Let $\sigma_i = \sqrt{|\lambda_i|}$, we can define the singular value decomposition of the operators T and T^* by*

$$T = \sum_{i=1}^{m} \sigma_i u_i \otimes v_i \text{ and } T^* = \sum_{i=1}^{m} \sigma_i v_i \otimes u_i,$$

which gives us the pair

$$T^* u_i = \sigma_i v_i \text{ and } T v_i = \sigma_i u_i \langle v_i, v_i\rangle_{\mathcal{K}}.$$

Proof We can show this by just substitution of the definitions. ■

Note that this is almost exactly like the Hilbertian case, except the difference in sign due to the term $\langle v_i, v_i \rangle_{\mathcal{K}}$.

Recall that in Hilbert space, the minimal norm solution to the interpolation problem is the projection of 0 onto the set of solutions $\{f \in \mathcal{H} \text{ such that } Tf = y\}$. Using this notion of projection, we can show the specific representation of the functions in $\mathcal{K}$ in terms of their eigenvalues and eigenvectors. The following proposition shows the form of the pseudo-inverse $f = T^{\dagger} y$ in Equation (4.9), and Equation (4.10) explicitly calculates the coefficients in the representer theorem.

Proposition 32 *Let f be a function in $\mathcal{K}$, a reproducing kernel Kreĭn space. The orthogonal projection of the origin onto the set of interpolation functions $\{Tf = y\}$ can be written (in the notation above) as*

$$f = T^* \sum_{i=1}^{m} \frac{\langle y, u_i \rangle}{\lambda_i} u_i \tag{4.9}$$

or

$$f = \sum_{j=1}^{m} \alpha_j k(x_j, \cdot) \text{ where } \alpha_j = \sum_{i=1}^{m} \frac{\langle y, u_i \rangle}{\lambda_i} u_i \tag{4.10}$$

Proof Since $\{v_1, \ldots, v_m\}$ forms an orthonormal system for $\mathcal{K}$, any function $f \in \mathcal{K}$ can be written as $f = \sum_{i=1}^{m} \beta_i v_i + g$, where $g \in N(T)$. Hence,

$$Tf = T\left(\sum_{i=1}^{m} \beta_i v_i + g\right) = \sum_{i=1}^{m} \beta_i T v_i = \sum_{i=1}^{m} \beta_i \frac{\lambda_i}{\sqrt{|\lambda_i|}} u_i.$$

We compute the coefficients β_i using the relationship $Tf = y$, and the spectral decomposition of $y = \sum_{i=1}^{m} \langle y, u_i \rangle u_i$. We obtain

$$\beta_i = \frac{\sqrt{|\lambda_i|}}{\lambda_i} \langle y, u_i \rangle.$$

Since f is the projection of the origin onto the interpolating set, then $g = 0$. Using the above value for β_i and the definition of v_i, we obtain the function f as a linear combination of kernels.

$$\begin{aligned} f &= \sum_{i=1}^{m} \frac{\sqrt{|\lambda_i|}}{\lambda_i} \langle y, u_i \rangle v_i \\ &= \sum_{i=1}^{m} \frac{\langle y, u_i \rangle}{\lambda_i} T^* u_i \\ &= T^* \sum_{i=1}^{m} \frac{\langle y, u_i \rangle}{\lambda_i} u_i \\ &= \sum_{j=1}^{m} \left(\sum_{i=1}^{m} \frac{\langle y, u_i \rangle}{\lambda_i} u_i \right)_j k(x_j, \cdot) \end{aligned}$$

Equation (4.9) shows that $f = T^*(TT^*)^{-1}y = T^\dagger y$, and Equation (4.10) details the form of the coefficients in the linear combination. ■

4.6 Summary of Learning in Kreĭn spaces

The aim of this chapter is to introduce the concept of an indefinite kernel to the machine learning community. These kernels, which induce an RKKS, exhibit many of the properties of positive semidefinite kernels. Several examples of indefinite kernels are given, along with their spectral properties. Due to the lack of positivity, we stabilize the loss functional instead of minimizing it. We have proved that stabilization provides us with a representer theorem, and also generalization error bounds via the Rademacher average. To demonstrate the difference between learning in an RKHS and learning in an RKKS, we showed the behaviour of the evaluation operator in terms of its spectrum.

We considered the regression problem in terms of the spline smoothing model, and showed that it may fail under certain conditions. In the following chapters, we propose a different approach to regularization of the linear system by using Krylov subspace methods.

Chapter 5

Regularization by Early Stopping

This chapter describes regularization of linear ill-posed problems by early stopping of conjugate gradient type algorithms. These algorithms provide a different class of methods from the current machine learning toolbox of constrained optimization problems and are introduced in Section 5.2. It can be applied to kernel methods in general, and importantly, it applies to problems where the kernel is indefinite. First, we analyse the regularization behaviour of the algorithm as a filter acting on the spectrum of the kernel matrix (Section 5.3). Second we demonstrate that for the Minimal Residual algorithm, the solution initially converges to the optimal and subsequently diverges when the number of iterations increases. This is the notion of semi-convergence (Section 5.4).

5.1 What is Regularization?

Intuitively, we want to have "smooth" solutions to our optimization problem. Regularization techniques aim to ensure the smoothness of the solution by augmenting the problem we want to solve. This was formalized by Tikhonov and Arsenin [1977], where regularization was defined in terms of solutions of ill-posed equations. Tikhonov proposed that in addition to minimizing the empirical risk, a stabilizing term is included in the minimization problem. The proposed solution has been successfully applied in many areas, especially for solving inverse problems [Wahba, 1980, Groetsch, 1984].

A second approach relates the solution of a linear equation to the optimization of the variational formulation [Vapnik, 1982, Chapter 2], and solving this using iterative methods. The regularization parameter in this case is the stopping index of the iterative method. This chapter investigates this approach to regularization in machine learning. We focus on conjugate gradient type methods which are also known as Krylov subspace methods. First, we define the idea of ill-posedness in the setting of kernel methods.

Let $T : \mathcal{K} \to \mathbb{R}^m$ be the evaluation functional where $f \mapsto Tf = [f(x_1), \ldots, f(x_m)]^\top$, that is T maps the function f from the reproducing kernel space (RKS) $\mathcal{K}$ to the vector consisting of the evaluation of this function on the input data $x_i \in \mathcal{X}$ for $i = 1, \ldots, m$.

$\mathcal{K}$ can be a Hilbert space or a Kreĭn space, and $\mathbb{R}^m$ has the standard Euclidean topology. We investigate the idea of regularization for a linear operator equation

$$Tf = y,$$

where $f \in \mathcal{K}$ and $y \in \mathcal{Y} = \mathbb{R}^m$.

Definition 33 (Linear Ill-Posed Problem) *Let $T : \mathcal{K} \to \mathcal{Y}$ be a linear operator. Then the equation*

$$Tf = y$$

where $f \in \mathcal{K}$ and $y \in \mathcal{Y}$ is

- well posed *if T is bijective and $T^\dagger$ is continuous, and*
- *it is* ill posed *otherwise.*

A possible approach to solve ill posed problems is to solve a series of approximate problems which are well posed. Furthermore, as the regularization parameter is reduced, the approximation approaches the original problem. Such methods are called regularization methods.

Definition 34 (Regularization Method) *Let $T : \mathcal{K} \to \mathcal{Y}$ be an injective bounded linear operator. Then an operator $T_\gamma^\dagger : \mathcal{Y} \to \mathcal{K}$, where $\gamma \in \mathbb{R}$ is the regularization parameter, and*

$$\lim_{\gamma \to 0} T_\gamma^\dagger T f = f, \text{ for all } f \in \mathcal{K},$$

is called a regularization method *for the operator T.*

The problem is compounded by the fact that in many applications, we do not have the exact labels y but a perturbed y^δ for some error level δ such that

$$\|y^\delta - y\| \leqslant \delta.$$

Note that we are not interested in $T^\dagger y^\delta$, even if it exists, but we are interested in the true value $T^\dagger y$, where $T^\dagger$ is the Moore-Penrose pseudo inverse of T. Hence we are interested in a regularization method which gives us

$$T_\gamma^\dagger y^\delta \simeq T^\dagger y.$$

Recall from Section 4.3.1 that we can express the linear operator equation as a set of linear equations

$$K\alpha = y.$$

Observe that there are several differences between the kernel methods setting and the solution of ill posed operator equations. First, since we are operating in reproducing kernel spaces, we have a finite dimensional problem as a result of Proposition 24. Second, even though we know that T is continuous (since $\mathcal{K}$ is a RKS), the labels y^δ may not be in the range of T. Third, the order optimal stopping rules (for example Morozov's discrepancy principle [Morozov, 1984]) do not apply since we do not know the noise level δ beforehand.

The idea behind regularization by early stopping is to find an approximation to the solution in a small subspace $\mathcal{S}$ of the possible solution space. That is, we are solving the following optimization problem,

$$\begin{array}{ll} \underset{\alpha \in \mathbb{R}^m}{\text{minimize}} & \|K\alpha - y\|^2 \\ \text{subject to} & \alpha \in \mathcal{S}. \end{array} \tag{5.1}$$

We are looking for the least squares interpolation of the given data from a subspace of the possible solution space. Observe that the regularization operator defined by this subspace approach is a projection operator, that is $T_\gamma^\dagger = \pi_\mathcal{S}$ where, $\pi_\mathcal{S}$ is a projection onto the subspace $\mathcal{S}$. We describe several different ways of choosing the subspace $\mathcal{S}$. The aim is to choose $\mathcal{S}$ such that

$$\pi_\mathcal{S} T^\dagger y^\delta \simeq T^\dagger y.$$

If we use the eigenvectors of K as the basis for our solution space, filtering the spectrum of K appropriately will select the subspace which is most useful for the problem. For more information on regularization methods, the reader is directed to surveys such as Girosi et al. [1995], Chen and Haykin [2002] and Engl and Kügler [2003].

5.2 Krylov Subspace Algorithms

The Conjugate Gradient method [Hestenes and Stiefel, 1952] is an effective method for solving linear symmetric positive definite systems. It is an iterative method that computes vector sequences of iterates (that is, successive approximations to the solution). The class of algorithms where the iterates are computed from the conjugated gradients are called conjugate gradient type algorithms. The sequence of iterates span a Krylov subspace, and hence such algorithms are also called Krylov subspace algorithms. A very readable tutorial, which motivates conjugate gradient algorithms from first principles, is Shewchuk [1994] and a historical survey of Krylov subspace methods can be found in van der Vorst [2000].

There are two major components to a linear conjugate gradient type algorithm:

- The computation of an optimum step size that minimizes the cost along a given direction, and

- the construction of an orthogonal basis for the solution space, the Krylov subspace.

It is known that these methods are equivalent to Lanczos methods [Lanczos, 1950, 1952], in the sense that they both find a solution in a sequence of Krylov subspaces. For an example of a derivation of the equivalence, see Golub and van Loan [1996]. The difference lies in the method used to construct the orthogonal basis. There have been numerous generalizations of the original algorithm, from which we choose several which are designed for linear symmetric indefinite problems. For more background on iterative methods in general, the reader is referred to books such as Axelsson [1994], Greenbaum [1997], Saad [2000] and Weiss [1996], and surveys such as Hanke and Hansen [1993], Engl and Kügler [2003] and Engl [2003]. There were recent advances in alternative iterative approaches, that do not use Krylov subspace iterations, for solving linear systems, including Calvetti and Reichel [2003, 2002], Calvetti et al. [1998], Hanke and Groetsch [1998] and Frankenberger and Hanke [2000]. We shall not discuss them here.

5.2.1 Iteration and Residual Polynomials

We analyse Krylov subspace methods that solve a linear system of equations

$$Ax = b,$$

where $A \in \mathbb{R}^{m\times m}$ a square symmetric matrix, and $x, b \in \mathbb{R}^m$. Observe that A can be an indefinite matrix. At iteration k, a Krylov subspace method finds a solution x_k which is an approximation of the true solution x from the Krylov space $x_0 + \mathcal{S}_k(b - Ax_0; A)$, where x_0 is some initial guess of the solution and the kth Krylov subspace is defined as

$$\mathcal{S}_k(z; G) = \text{span}\{z, Gz, G^2z, \ldots, G^{k-1}z\}.$$

The iterates x_k of a Krylov subspace method can be expressed in terms of the iteration polynomial. Let Π_k be the space of polynomials of degree k where $\Pi_{-1} = \{0\}$, and Π_k^0 the set of normalized polynomials of degree k that is $\Pi_k^0 = \{p \in \Pi_k | p(0) = 1\}$. The coefficients of a polynomial $q_{k-1} \in \Pi_{k-1}$ can be found such that the iterate at step k is given by

$$x_k = x_0 + q_{k-1}(A)(b - Ax_0).$$

The residual at iteration k, is defined to be $r_k := b - Ax_k$, and can be expressed in terms of another polynomial $p_k \in \Pi_k^0$ as

$$r_k = p_k(A)(b - Ax_0).$$

The iteration polynomials q_{k-1} and the residual polynomials p_k are related by

$$p_k(A) = \mathbf{I} - Aq_{k-1}(A). \tag{5.2}$$

Since A can be decomposed into its eigenvalues λ_i and eigenvectors u_i such that, $Au_i = \lambda_i u_i$ for $i = 1, \ldots, m$, any polynomial on the operator A can be seen as affecting each eigenvalue individually. The polynomials defined above can be equivalently expressed in terms of the eigenvalues of A. This equivalence will be useful for proving bounds for conjugate gradient type methods, and is derived in Appendix C.1.

Krylov subspace methods are efficient because of a cheap recursion for the computation of a new iterate x_{k+1} given the previous iterates $x_0, \ldots, x_k$. There are numbers $\alpha_k \neq 0$, and $\beta_k > 0$ such that for $k \geqslant 1$, the residual polynomials satisfy the recursion

$$p_0 = 1; \quad p_1 = 1 - \alpha_0\lambda; \quad p_{k+1} = -\alpha_k\lambda p_k + p_k - \alpha_k\frac{\beta_k}{\alpha_{k-1}}(p_{k-1} - p_k). \tag{5.3}$$

The associated iteration polynomials satisfy the recursion

$$q_{-1} = 0; \quad q_0 = \alpha_0; \quad q_k = q_{k-1} + \alpha_k\left(p_k + \frac{\beta_k}{\alpha_{k-1}}(q_{k-1} - q_{k-2})\right). \tag{5.4}$$

Hence the algorithm computes a new iterate x_{k+1} using only the two previous iterates x_k and x_{k-1}. This computation can be computed as one three term recursion (which gives rise to the Lanczos approach), or as a pair of two term recursions (which gives rise to the conjugate gradient approach).

5.2.2 Details of Conjugate Gradient Methods

The various approaches to Krylov subspace algorithms differ in which linear system they solve, the cost function that is being optimized, the matrix used to define conjugacy and the method used to generate the subsequent search directions. Table 5.1 summarizes the methods used and Figure 5.1 and 5.2 shows the implementation used in the experiments later in this thesis, which were based on Hanke [1995a].

Although the literature recommends using preconditioning [Benzi, 2002] with conjugate gradient methods, we did not use any preconditioning on our problems, as we wanted to analyse the regularization properties of the algorithm itself.

We show below that CG (Figure 5.1) generates residual polynomials and iteration

Definition	CG	MR	CGNE	MR-II
System	$Ax = b$	$Ax = b$	$A^2x = Ab$	$Ax = b$
Cost, $J(x)$	$\frac{1}{2}x^\top Ax - b^\top x$	$\frac{1}{2}\\|Ax - b\\|^2$	$\frac{1}{2}x^\top A^2x - b^\top Ax$	$\frac{1}{2}\\|Ax - b\\|^2$
Orthogonality	$d_i^\top Ad_j = 0$	$d_i^\top A^2d_j = 0$	$d_i^\top A^2d_j = 0$	$d_i^\top A^2d_j = 0$
	$r_i^\top r_j = 0$	$r_i^\top Ar_j = 0$	$r_i^\top A^2r_j = 0$	
New Direction, d_k	$r_k + \beta_kd_{k-1}$	$r_k + \beta_kd_{k-1}$	$Ar_k + \beta_kd_{k-1}$	

Table 5.1: Summary of definitions used in Krylov Subspace Methods

```
$k = 0;\ d_0 = r_0 = b - Ax_0;$
while (not stop) do
    $\alpha_k = \frac{r_k^\top r_k}{d_k^\top Ad_k};$
    $x_{k+1} = x_k + \alpha_kd_k;$
    $r_{k+1} = r_k - \alpha_kAd_k;$
    $\beta_{k+1} = \frac{r_{k+1}^\top r_{k+1}}{r_k^\top r_k}$
    $d_{k+1} = r_{k+1} + \beta_{k+1}d_k$
end while
```

```
$k = 0;\ d_0 = r_0 = b - Ax_0;$
while (not stop) do
    $\alpha_k = \frac{r_k^\top Ar_k}{d_k^\top A^2d_k};$
    $x_{k+1} = x_k + \alpha_kd_k;$
    $r_{k+1} = r_k - \alpha_kAd_k;$
    $\beta_{k+1} = \frac{r_{k+1}^\top Ar_{k+1}}{r_k^\top Ar_k}$
    $d_{k+1} = r_{k+1} + \beta_{k+1}d_k$
end while
```

Figure 5.1: Left: Conjugate Gradient algorithm (CG), Right: Minimal Residual algorithm (MR)

```
$k = 0;\ d_0 = r_0 = b - Ax_0;$
$d_0 = Tr_0;$
while (not stop) do
    $\alpha_k = \frac{r_k^\top A^2r_k}{d_k^\top A^2d_k};$
    $x_{k+1} = x_k + \alpha_kd_k;$
    $r_{k+1} = r_k - \alpha_kAd_k;$
    $\beta_{k+1} = \frac{r_{k+1}^\top A^2r_{k+1}}{r_k^\top A^2r_k}$
    $d_{k+1} = Ar_{k+1} + \beta_{k+1}d_k$
end while
```

```
$r_0 = y - Sx_0;\ r_1 = r_0;\ x_1 = x_0;$
$v_{-1} = 0;\ v_0 = Sr_0;\ w_{-1} = 0;\ w_0 = Sv_0;$
$\beta = \|w_0\|;\ v_0 = v_0/\beta;\ w_0 = w_0/\beta;$
$k = 1;$
while (not stop) do
    $\varrho = \langle r_k, w_{k-1}\rangle;\ \alpha = \langle w_{k-1}, Sw_{k-1}\rangle;$
    $x_{k+1} = x_k + \varrho v_{k-1};\ r_{k+1} = r_k + \varrho w_{k-1};$
    $v_k = w_{k-1} - \alpha v_{k-1} - \beta v_{k-2};$
    $w_k = Sw_{k-1} - \alpha w_{k-1} - \beta w_{k-2};$
    $\beta = \|w_k\|;\ v_k = v_k/\beta;\ w_k = w_k/\beta;$
    $k = k + 1;$
end while
```

Figure 5.2: Left:Conjugate Gradient on Normal Equations algorithm (CGNE), Right: Lanczos based Minimal Residual Algorithm (MR-II)

polynomials that have recursions as shown in Equation (5.3) and (5.4) respectively. The derivations of the other algorithms are omitted as they are similar. The reason we require these recursions is because we analyse the regularization properties of the algorithm in terms of the iteration polynomials q_{k-1}.

Proposition 35 *The algorithm CG has residual polynomials and iteration polynomials that form three term recurrences as defined in Equation (5.3) and (5.4) respectively.*

Proof We show that the CG algorithm as defined in Figure 5.1 has a three term recursion for its residual polynomials. From the definition of the iterates $x_{k+1} = x_k + \alpha_k d_k$ and the definition of the search direction $d_{k+1} = r_{k+1} + \beta_{k+1} d_k$, we get the following calculation:

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k d_k \\ &= x_{k-1} + \alpha_k d_k + \alpha_{k-1} d_{k-1} \\ &= x_{k-1} + \alpha_k (r_k + \beta_k d_{k-1}) + \alpha_{k-1} d_{k-1}. \end{aligned}$$

Pre-multiplying by A on both sides,

$$Ax_{k+1} = Ax_{k-1} + \alpha_k Ar_k + \alpha_k \beta_k Ad_{k-1} + \alpha_{k-1} Ad_{k-1}.$$

Observe that $r_k - r_{k+1} = \alpha_k Ad_k$, therefore

$$Ax_{k+1} = Ax_{k-1} + \alpha_k Ar_k + \left(\tfrac{\alpha_k \beta_k}{\alpha_{k-1}} + 1\right)(r_{k-1} - r_k).$$

By definition of the residual, $Ax_{k+1} = b - r_{k+1}$, and hence

$$\begin{aligned} b - r_{k+1} &= b - r_{k-1} + \alpha_k Ar_k + \tfrac{\alpha_k \beta_k}{\alpha_{k-1}}(r_{k-1} - r_k) + r_{k-1} - r_k \\ r_{k+1} &= -\alpha_k Ar_k + r_k - \tfrac{\alpha_k \beta_k}{\alpha_{k-1}}(r_{k-1} - r_k). \end{aligned}$$

Associating the residuals with the residual polynomials,

$$p_{k+1} = -\alpha_k \lambda p_k + p_k - \tfrac{\alpha_k \beta_k}{\alpha_{k-1}}(p_{k-1} - p_k).$$

The iteration polynomials can be found by substituting Equation (5.2). ■

Many results on the convergence of Krylov subspace methods determines the behaviour of the algorithms when the error approaches zero or when the problem is well posed [Fischer et al., 1996, Hyvönen and Nevanlinna, 2000]. Since machine learning problems are inherently ill-posed, we cannot expect the solution to asymptotically converge to the optimal. In fact, if T is a compact operator, the inverse is unbounded.

In the finite dimensional case, the solution initially converges to the optimal, then as the number of iterations increases, begin to diverge away from the solution. Hence regularization is obtained by stopping the iterative procedure early, which means that the stopping index is the regularization parameter. Note that our work is more closely related to analysis that determine the optimal regularization parameter, rather than the asymptotic behaviour of Krylov subspace methods. For a survey of such methods, see Hanke and Hansen [1993], Hansen [1998] and Kilmer and O'Leary [2001].

5.2.3 Relation to Partial Least Squares

CGNE has been shown to solve the partial least squares (PLS) problem, where the number of dimensions in the PLS estimator is the stopping index of CGNE. This equivalence between PLS, Lanczos iterations and CGNE was used in Phatak and de Hoog [2002] to show several properties of PLS, including the fact that PLS is a shrinkage estimator [Hastie et al., 2001, Section 3.4.3]. Using alternative methods, de Jong [1995] and Goutis [1996] also proved the shrinkage properties of PLS. However, the coefficients of the estimator shrinks in some directions and expands in others [Lingjærde and Christophersen, 2000, Butler and Denham, 2000].

A second useful property of PLS is that it fits closer than principal components regression [de Jong, 1993, Phatak and de Hoog, 2002]. This implies that for the same level of approximation, PLS would require fewer dimensions than principal components regression. Due to its simple algorithm and efficient implementation, PLS has been very popular in the Chemometrics literature. Surveys of the properties of PLS include Helland [1988, 2001].

Our results provide an alternative view to the analysis of kernel PLS which was proposed by Rosipal and Trejo [2001] and Bennett and Embrechts [2003]. Note that the traditional algorithm used for PLS and kernel PLS, which is called NIPALS, is different from CGNE which we consider here. In addition, we observe similar behaviour for the other conjugate gradient type algorithms analyzed here.

5.3 Filter Functions

In this section, we analyze the effect of a particular choice of regularization method on the spectrum of the operator. Recall from Proposition 32 that we can express the solution of the interpolation problem as

$$f = T^* \sum_{i=1}^{m} \frac{\langle y, u_i \rangle}{\lambda_i} u_i.$$

We control the spectrum of the operator K via a real valued function φ on the eigenvalues λ_i. We define a filter function $\varphi(\lambda_i, \gamma)$ which is piecewise continuous for a regularization parameter $\gamma \neq 0$, and converges to 1 for all λ_i as $\gamma \to 0$. Hence the regularized solution [Engl and Kügler, 2003, Theorem 1 and 2] is

$$f_\gamma = \sum_{i=1}^{m} \varphi(\lambda_i, \gamma) \frac{\langle y, u_i \rangle}{\lambda_i} T^* u_i.$$

Using spectral mapping theorem, we can control the inverse operator $T^\dagger$ using the filter function since it filters the spectrum of the inverse operator. Observe that the filter function in this case only operates on the eigenvalues of the finite dimensional matrix K. Hence the regularization operator is defined by the filter function.

5.3.1 Truncated Spectral Factorization

Intuitively, we associate eigenvalues with large absolute values to the underlying function, and associate eigenvalues close to zero with signal noise. The Truncated Spectral Factorization (TSF) [Engl and Kügler, 2003] method can be obtained by setting all the eigenvalues of small magnitude to zero. This means that the solution is in the subspace

$$\mathcal{S} = \operatorname{span}\{T^* u_i\}, |\lambda_i| > \gamma,$$

and the filter function is given by

$$\varphi(\lambda_i, \gamma) = \begin{cases} 1 & |\lambda_i| \geqslant \gamma \\ 0 & |\lambda_i| < \gamma \end{cases}$$

The resulting regression algorithm when truncated spectral factorization is applied prior to performing regression is also known as principal components regression [Hastie et al., 2001, Section 3.4.4].

5.3.2 Tikhonov Regularization

The least squares approximation can be obtained by solving a special case of the optimization problem in the representer theorem, that is when we perform Tikhonov regularization of the spline interpolation problem (Section 4.3.2). This is equivalent to solving the linear system $(TT^* + \gamma \mathbf{I})\alpha = y$. Since $TT^* = K$, for values of the regularization parameter γ which equal a negative eigenvalue of the Gram matrix K, $(K + \gamma \mathbf{I})$ is singular. Note that in the case where K is positive, this does not occur. Hence, solving the Tikhonov regularization problem directly may not be successful.

Tikhonov regularization can be seen as setting

$$\varphi(\lambda_i, \gamma) = \frac{\lambda_i}{\lambda_i + \gamma}.$$

Alternatively, we can solve the linear system posed by applying Tikhonov regularization to the empirical feature map, $K : \mathbb{R}^m \to \mathbb{R}^m$ such that $\alpha \mapsto TT^*\alpha = K\alpha$. This is equivalent to solving normal equations $K^\top K\alpha = K^\top y$ with a regularization term, and is also known as weight decay or ridge regression [Hoerl and Kennard, 1970].

$$\min_\alpha \frac{1}{2}\|y - K\alpha\|^2 + \gamma\|K\alpha\|^2,$$

where the norms are in $\mathbb{R}^m$. Note that we are applying Tikhonov regularization with the regularization operator K, which gives us the linear system $(K^\top K + \gamma\mathbf{I})\alpha = K^\top y$. The matrix $K^\top K$ is positive, and therefore a unique solution exists for $\gamma > 0$. The corresponding filter function is given by

$$\varphi(\lambda_i, \gamma) = \frac{\lambda_i^2}{\lambda_i^2 + \gamma}.$$

5.3.3 Steepest Descent

Iterative methods can be used to minimize the squared error $J(f) := \frac{1}{2}\|Tf - y\|^2$. Since $J(f)$ is convex, we can perform gradient descent. Since $\nabla_f J(f) = T^*Tf - T^*y$, we have the iterative definition $f_{k+1} = f_k - \gamma(T^*Tf - T^*y)$, which results in Landweber-Fridman (LF) iteration [Hanke and Hansen, 1993]. This has the filter function

$$\varphi(\lambda_i, \gamma, k) = 1 - (1 - \gamma\lambda_i)^k,$$

where k is the number of iterations. The solution subspace in this case is the polynomial

$$\mathcal{S} = \operatorname{span}\{(\mathbf{I} - \gamma T^*T)^k T^*y\} \text{ for } 1 \leqslant k \leqslant m.$$

This comes from a class of iterative methods called stationary iterative methods, where the coefficients of iteration do not vary. This is also known as Richardson iteration or Picard's iteration. The regularization behaviour of this algorithm was previously analysed in Wahba [1987]. Examples of the four filter functions described so far are plotted in Figure 5.3.

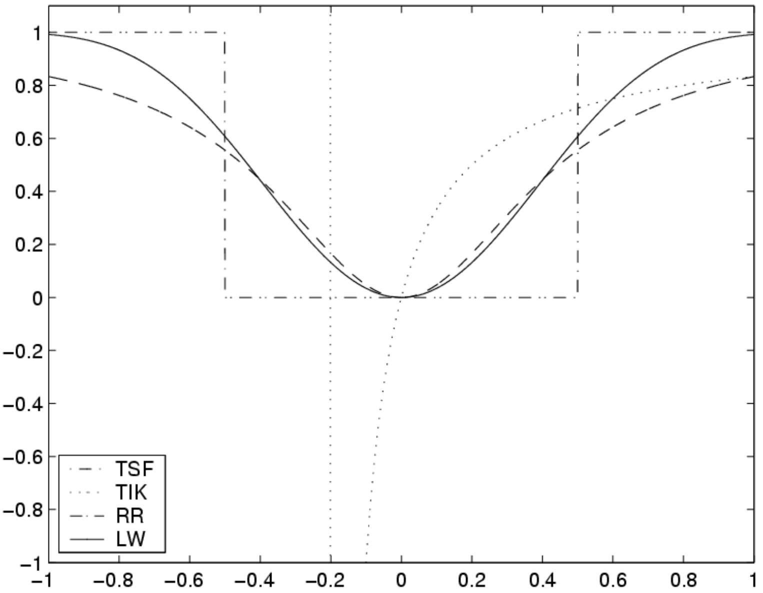


Figure 5.3: Filter functions for truncated spectral regularization (TSF) ($\gamma = 0.5$), Tikhonov Regularization (TIK) ($\gamma = 5$) Ridge Regression (RR) ($\gamma = 5$) and Landweber Iteration (LW) ($\gamma = 0.2, k = 7$)

5.3.4 Krylov Subspace Methods

Recall from Section 5.2 that the conjugate gradient type methods finds solutions which can be expressed in terms of the iteration polynomial q_{k-1}. Setting $\alpha_0 = 0$, we can derive the filter function associated with the algorithm as follows:

$$\begin{aligned} f_k &= T^* \alpha_k \\ &= T^* q_{k-1}(K) y \\ &= T^* q_{k-1}(U^\top \Lambda U) y \\ &= \sum_{i=1}^{m} q_{k-1}(\lambda_i) \langle y, u_i \rangle T^* u_i \\ &= \sum_{i=1}^{m} \lambda_i q_{k-1}(\lambda_i) \frac{\langle y, u_i \rangle}{\lambda_i} T^* u_i. \end{aligned}$$

Therefore the filter function is

$$\varphi(\lambda_i, k) = \lambda_i q_{k-1}(\lambda_i).$$

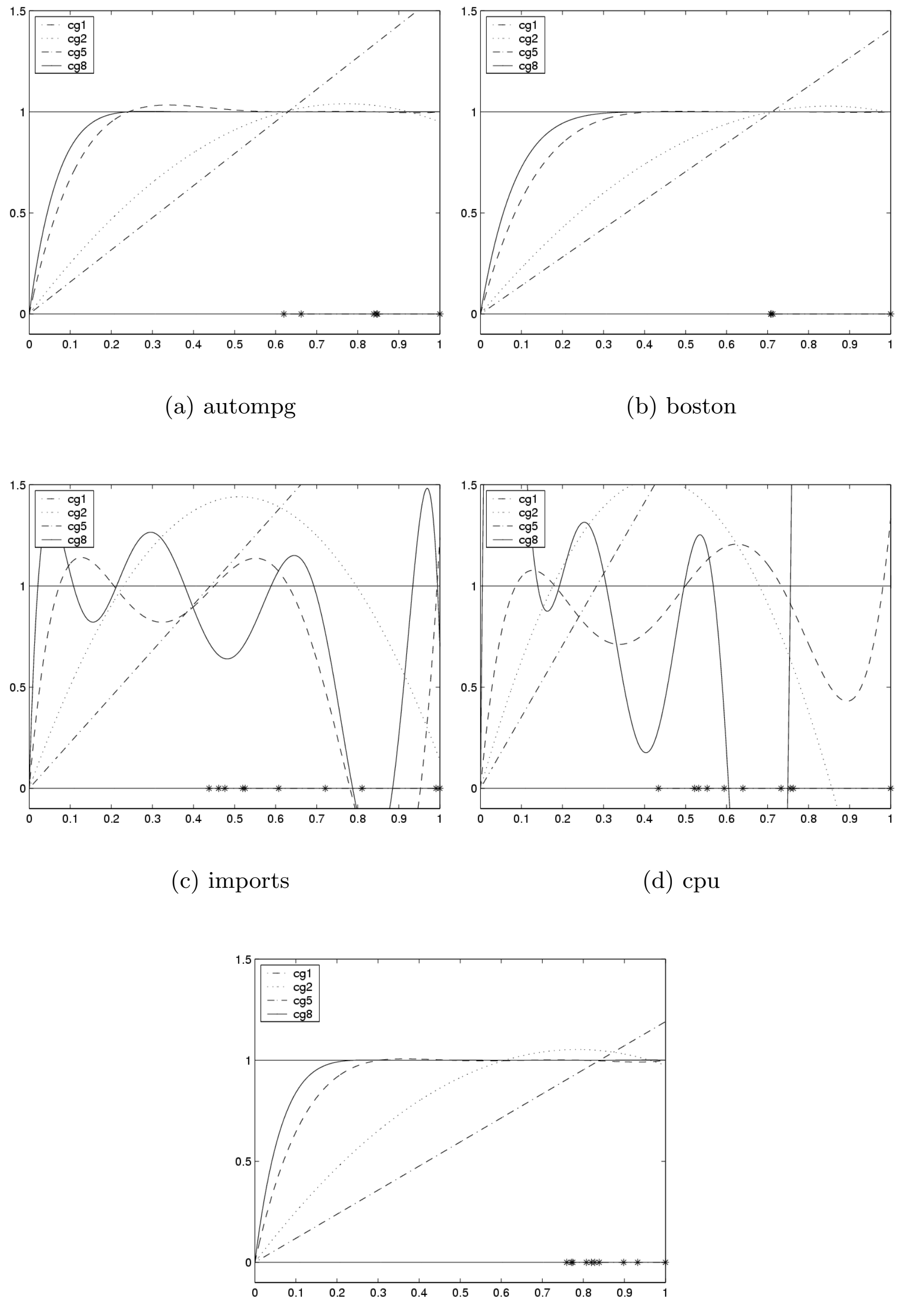
Note that the filter functions are data dependent since the coefficients of the iteration polynomials are dependent on y. Filter functions for Krylov subspace algorithms have been investigated in Haber [1997, Chapter 4]. Examples of filter functions generated by CG, MR, CGNE and MR-II on UCI datasets using the Gaussian kernel are shown in Figure 5.4, 5.5, 5.6 and 5.7 respectively. Examples of filter functions generated by MR, CGNE and MR-II on UCI datasets using the Epanechnikov kernel are shown in Figure 5.8, 5.9 and 5.10 respectively. The kernel matrices are normalized by its norm, and hence for the positive semidefinite Gaussian kernel, the eigenvalues are in the interval $[0, 1]$.

Compared with Figure 5.3, the filter functions for Krylov methods are "sharper" than the fixed filter functions. Compared with TSF, it sets more eigenvalues to 1 with only a few iterations of conjugate gradient type algorithms. This result is in agreement with the fact pointed out in Section 5.2.3 that PLS needs fewer dimensions that PCR. The observations that PLS shrinks in some directions and expands in others are also apparent from the filter functions, as there are sections above 1, which implies that the filter expands the spectrum in the direction of the eigenvector concerned.

5.4 Semi-convergence of Minimal Residual

We analyse MR (Figure 5.1) in more detail when K is positive definite. Recall that we would like to solve the linear equation $K\alpha = y$, but we only have the noisy right hand side y^δ. MR minimises the residual norm $\|r_k\| = \|y - K\alpha_k\|$ among all the possible values of α in the Krylov subspace $\mathcal{S}_k$. For $y \in \mathcal{R}(K)$, the iterates of MR converge to $K^\dagger y$ asymptotically [Hanke, 1995a, Theorem 3.4]. However, in the presence of noise, we cannot expect y^δ to be still in the range of K. In this case, the norm of the residual diverges [Hanke, 1995a, Theorem 3.5]. Note that this analysis holds true for general positive semidefinite operators, not just matrices. Even in the finite case, the norm of the error $\|\alpha_k\|$ for a noisy right hand side y^δ does not decrease monotonically with the number of iterations. In fact, trying to reduce the residual norm below a certain level causes large increases in the error [Kilmer and Stewart, 1999, Theorem 3.1]. This phenomenon of the error initially decreasing then subsequently increasing is called semi-convergence [Natterer, 1982]. Hence the stopping index of the conjugate gradient algorithm is the regularization parameter. An example of a method to choose this parameter is Morozov's discrepancy principal that was introduced in Morozov [1984]. However, as mentioned earlier, since we do not have prior knowledge about the noise level δ, we cannot use this approach.

The regularization properties of CGNE was studied in Nemirovskii [1986], where the discrepancy principle was suggested as the stopping criteria, and order-optimal bounds was established for the approximations. A variant of the problem where CG is applied directly to $AA^\top w = b$ where $x = A^\top w$ was called the Minimal Error method. The appropriate stopping rule was investigated in Hanke [1995b]. Previous work on the regularization properties of MR is in Kilmer and Stewart [1999], and the regularization



(a) autompg

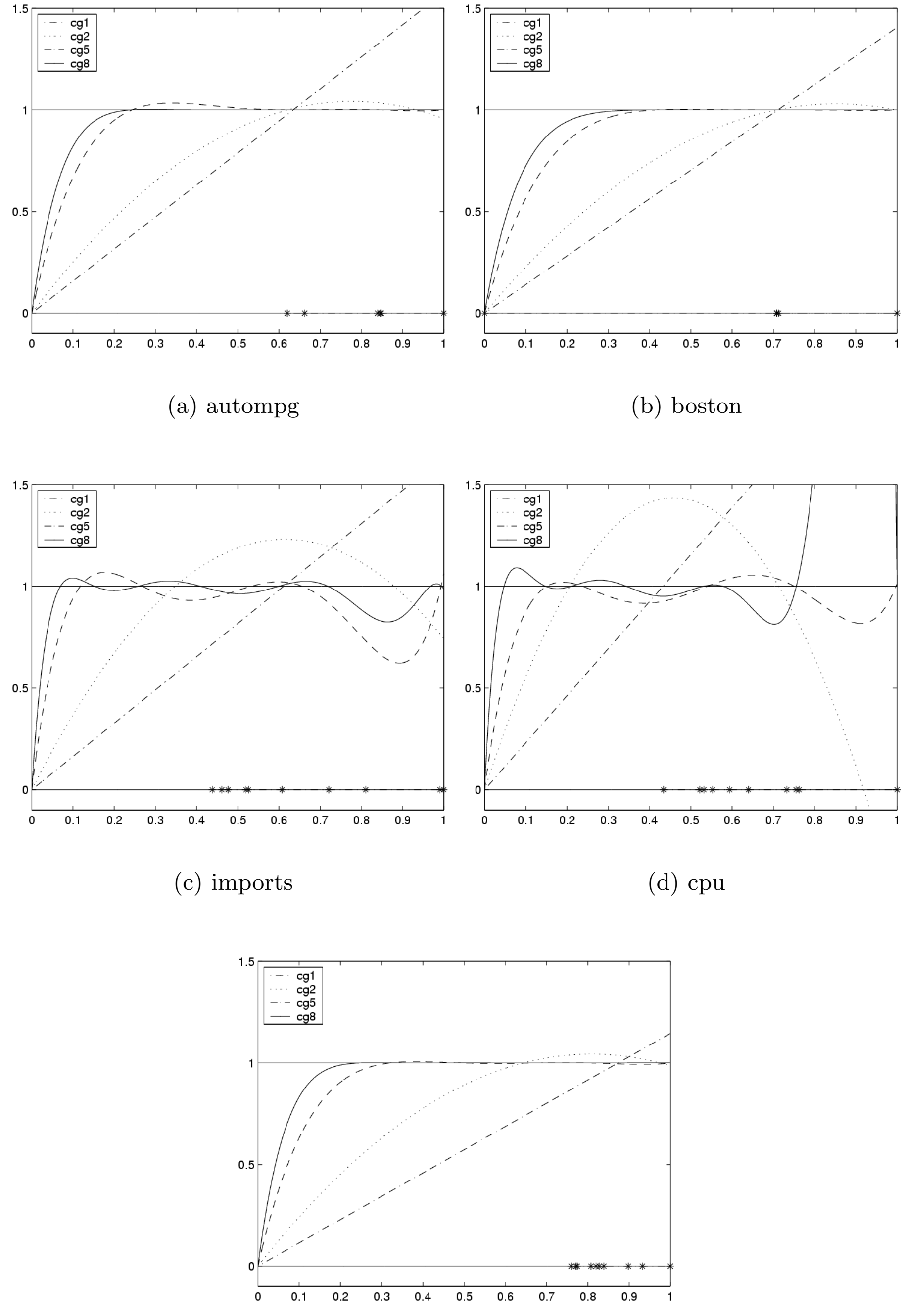
(b) boston

(c) imports

(d) cpu

(e) servo

Figure 5.4: Filter functions for CG using the Gaussian kernel on the various datasets after 1, 2, 5 and 8 iterations (denoted by cg1, cg2, cg5 and cg8 respectively). The asterisks on the horizontal axis indicates the location of the 10 principal eigenvalues of K.



(a) autompg

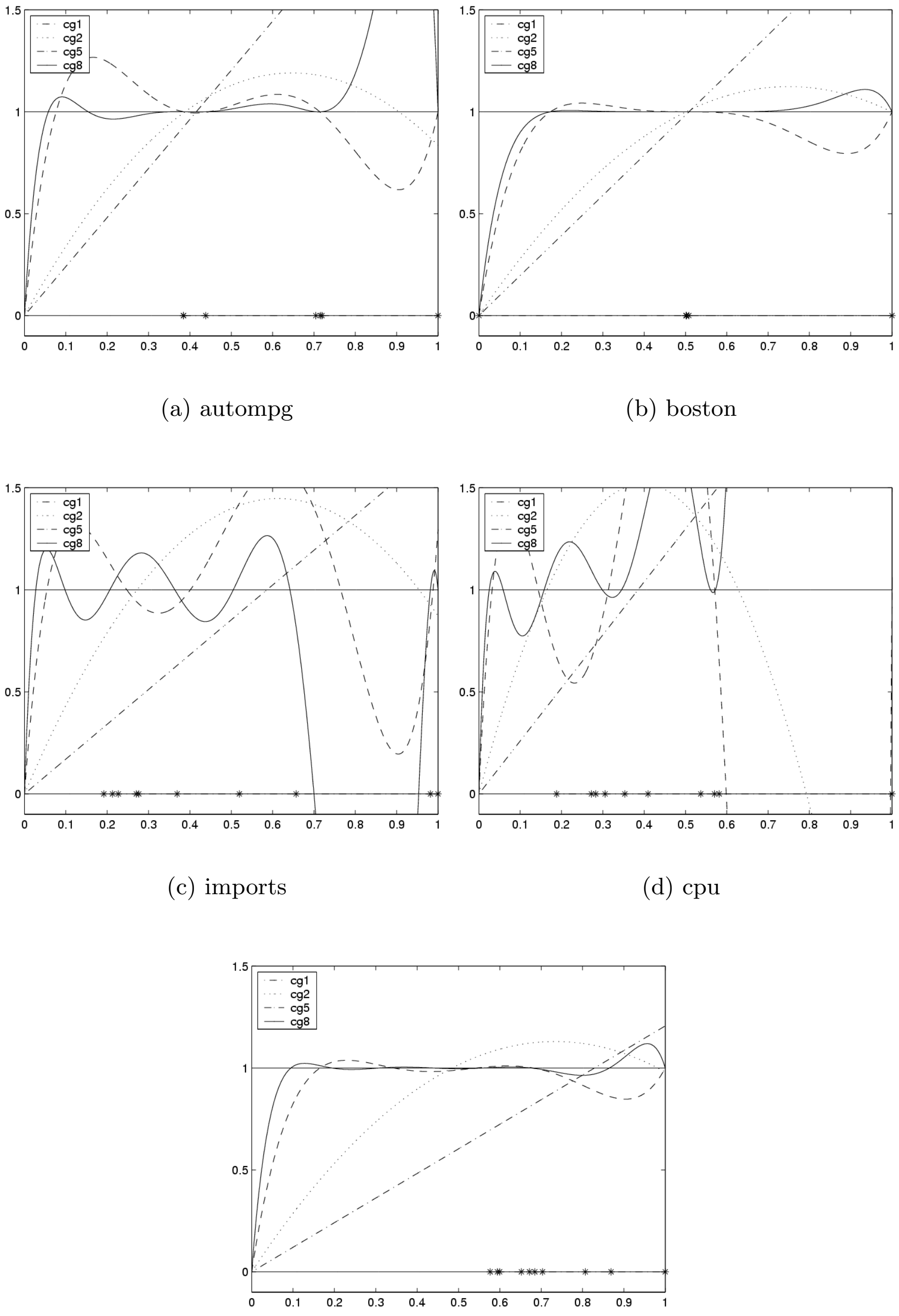
(b) boston

(c) imports

(d) cpu

(e) servo

Figure 5.5: Filter functions for MR using the Gaussian kernel on the various datasets after 1, 2, 5 and 8 iterations (denoted by cg1, cg2, cg5 and cg8 respectively). The asterisks on the horizontal axis indicates the location of the 10 principal eigenvalues of K.



(a) autompg

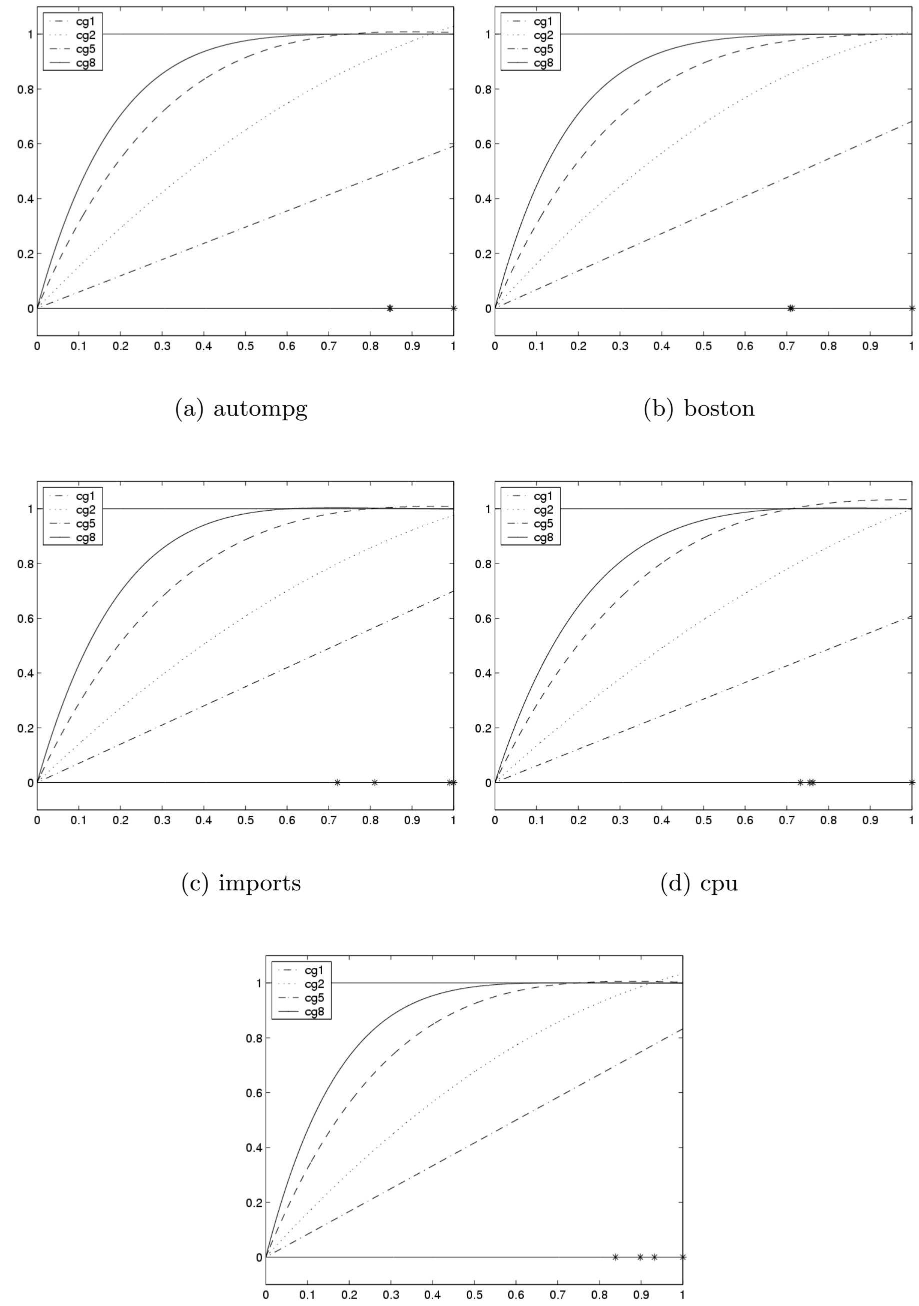
(b) boston

(c) imports

(d) cpu

(e) servo

Figure 5.6: Filter functions for CGNE using the Gaussian kernel on the various datasets after 1, 2, 5 and 8 iterations (denoted by cg1, cg2, cg5 and cg8 respectively). The asterisks on the horizontal axis indicates the location of the 10 principal eigenvalues of K^2.



(a) autompg

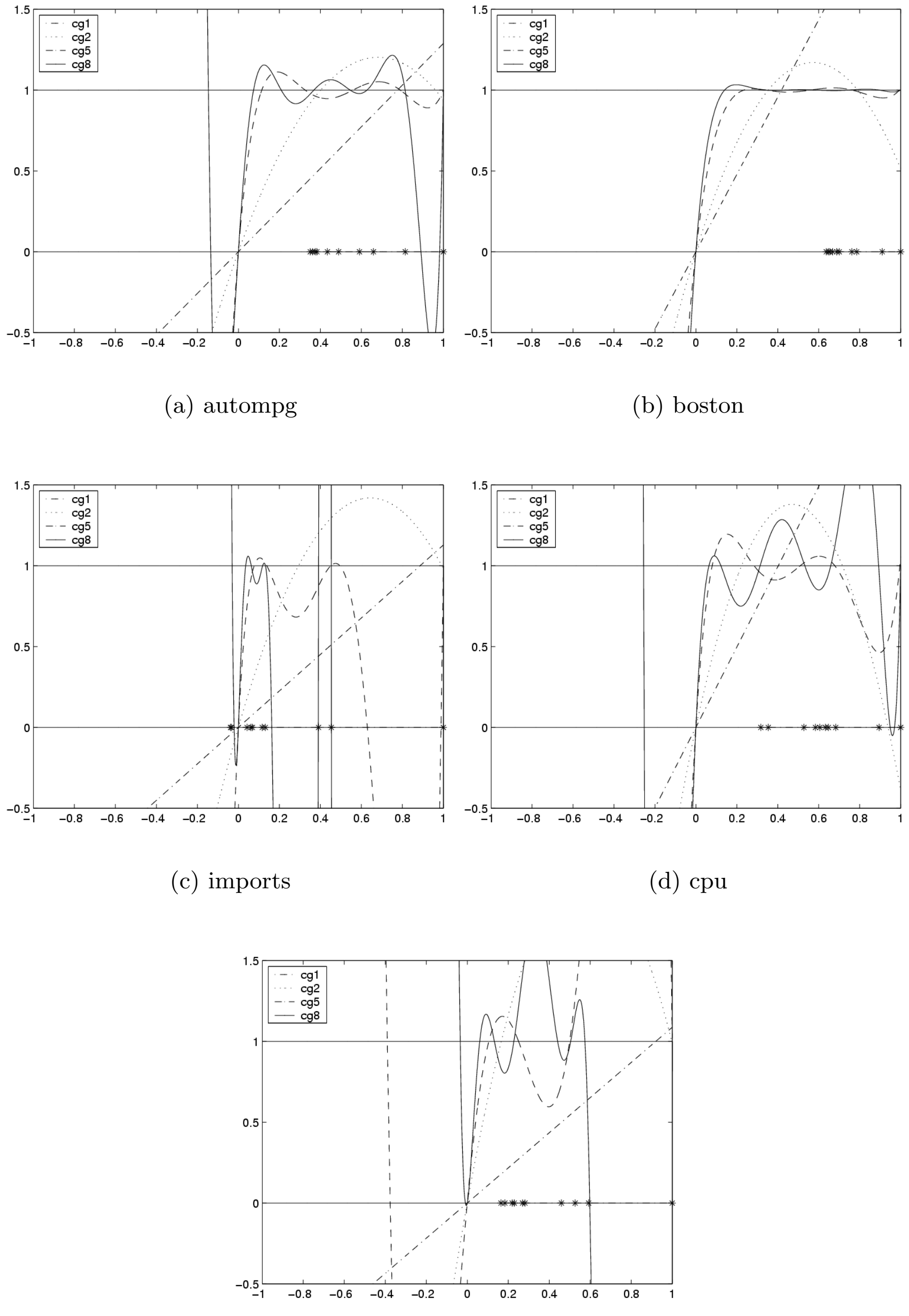
(b) boston

(c) imports

(d) cpu

(e) servo

Figure 5.7: Filter functions for MR-II using the Gaussian kernel on the various datasets after 1, 2, 5 and 8 iterations (denoted by cg1, cg2, cg5 and cg8 respectively). The asterisks on the horizontal axis indicates the location of the 10 principal eigenvalues of K.



(a) autompg

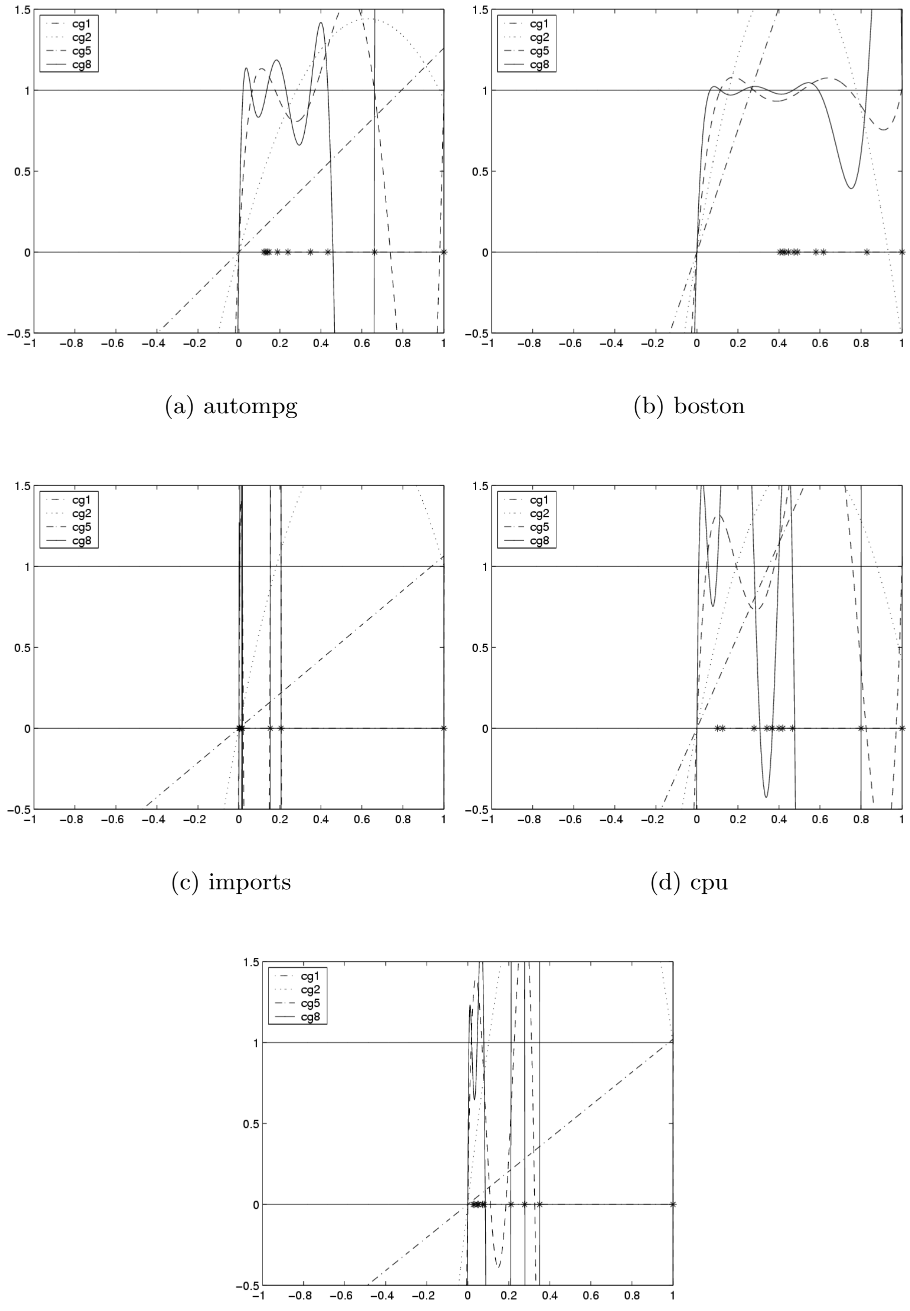
(b) boston

(c) imports

(d) cpu

(e) servo

Figure 5.8: Filter functions for MR using the Epanechnikov kernel on the various datasets after 1, 2, 5 and 8 iterations (denoted by cg1, cg2, cg5 and cg8 respectively). The asterisks on the horizontal axis indicates the location of the 10 principal eigenvalues of K.



(a) autompg

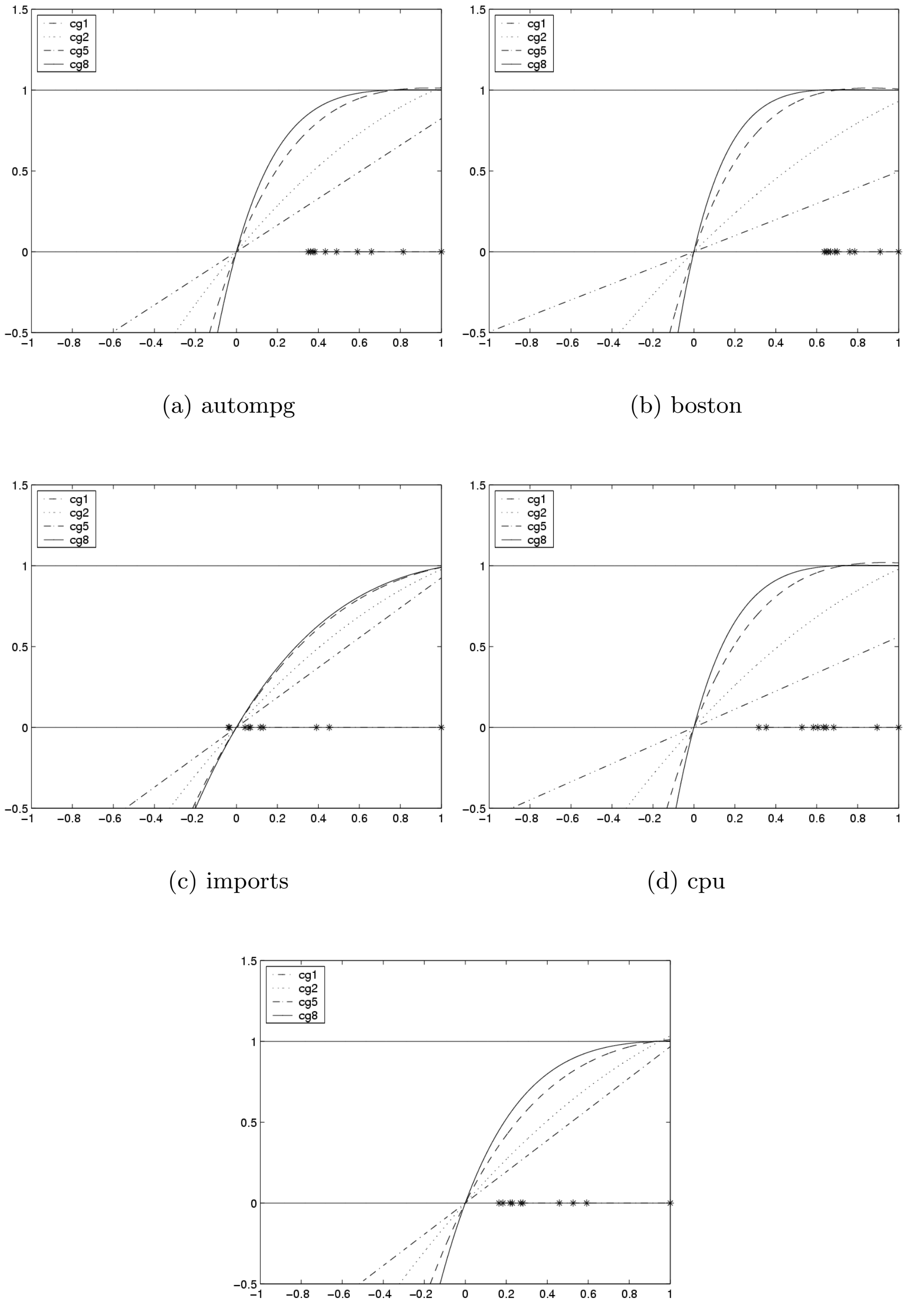
(b) boston

(c) imports

(d) cpu

(e) servo

Figure 5.9: Filter functions for CGNE using the Epanechnikov kernel on the various datasets after 1, 2, 5 and 8 iterations (denoted by cg1, cg2, cg5 and cg8 respectively). The asterisks on the horizontal axis indicates the location of the 10 principal eigenvalues of K^2.



(a) autompg

(b) boston

(c) imports

(d) cpu

(e) servo

Figure 5.10: Filter functions for MR-II using the Epanechnikov kernel on the various datasets after 1, 2, 5 and 8 iterations (denoted by cg1, cg2, cg5 and cg8 respectively). The asterisks on the horizontal axis indicates the location of the 10 principal eigenvalues of K.

properties of MR-II is investigated in Hanke [1995a, Chapter 6].

In view of these results, we would like to derive an expression for the expected risk. Figure 5.11 shows a summary of the notation used in this section. We denote the iterates generated by MR on the noisy problem by α_k^δ. Hence for a particular K and y^δ, MR generates a sequence $\alpha_0^\delta, \dots, \alpha_k^\delta, \dots, \alpha_m^\delta$. Associated with each iterate α_k^δ is a function defined by $f_k = T^* \alpha_k^\delta$. We denote the underlying target function generating the data by t, and the projection of t onto the range of T^* by t^*. The empirical risk $R_{\text{emp}}(f)$ is the residual norm for a particular choice of α, that is $R_{\text{emp}}(f_k) = \|y - K\alpha_k\|$. The aim of the following theorems is to qualitatively relate the empirical error $R_{\text{emp}}(f_k) - R_{\text{emp}}(t)$, and the expected error $R(f_k) - R(t)$, where $R(f) := \sqrt{\int (f(x) - y)^2 dP(x, y)}$.

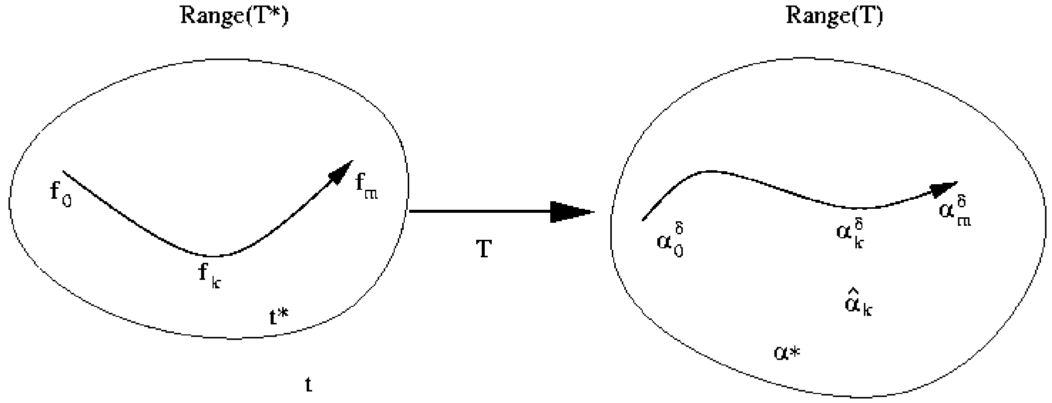


Figure 5.11: A guide to the notation used in Section 5.4. For a noisy problem $K\alpha = y^\delta$, we denote the iterates of MR by $\alpha_0^\delta, \dots, \alpha_k^\delta$ where k is the number of iterations. A function $f_k := T^* \alpha_k^\delta$ is associated with each iterate α_k^δ. We denote the true underlying function by t and its projection onto the range of T^* as t^*. Associated with the ideal function is a coefficient vector α^*.

The analysis of conjugate gradient type algorithms is more complex than those for the regularization methods described in Section 5.3.1, 5.3.2 and 5.3.3. This is because the regularization operator is non-linear as it involves both K and y^δ. Let α_k^δ be the iterates corresponding to the solution of the interpolation equation with noise, that is $K\alpha^\delta = y^\delta$. As can be observed by the figures in Section 5.3.4, the filters and hence the regularization properties are dependent on the slope of $p_k(\lambda)$ at the origin. Let $\theta_{i,k}$ be the ith root of the residual polynomial $p_k(\lambda)$ of order k. Recall that we can express a polynomial as a product of its roots, that is

$$p_k(\lambda) = \prod_{i=1}^{k} \left(1 - \frac{\lambda}{\theta_{i,k}}\right) \quad \text{and hence } p_k'(0) = -\sum_{i=1}^{k} \frac{1}{\theta_{i,k}}.$$

Theorem 36 provides the relationship between the empirical risk on the data and the best possible empirical risk. The proof follows closely that of Hanke [1995a, Lemma 3.7], where the norms of the residuals are investigated. As the proof is not well known in the machine learning community, it is reproduced in a modified form in Appendix C.2. The proof of Theorem 36 involves four technical lemmas about the residual polynomials $p_k(\lambda)$ which are also proved in Appendix C.2.

Theorem 36 *Let α^* be the solution of the underlying problem $K\alpha = y$, and α_k^δ be the iterates of MR applied to the noisy problem $K\alpha = y^\delta$. If $y \in \mathcal{R}(K)$, then*

$$R_{\text{emp}}(f_k) - R_{\text{emp}}(t) \leqslant \frac{2}{|p_k'(0)|}\|\alpha^*\|,$$

where $f_k = T^*\alpha_k^\delta$, and t is the optimal function associated with the noiseless data y, that is $y = Tt$.

Theorem 36 shows that the empirical risk is not too far from the empirical risk achieved by the best function. Since $|p_k'(0)| \geqslant k$, the gap narrows with the number of iterations. The gradient of the residual at zero $p_k'(0)$ is also the quantity of interest that determines the rate of convergence and divergence of the error. Based on previous work by other researchers such as Hanke and Hansen [1993], Hansen [1998], we do not expect the real error to converge. The phenomenon of semi-convergence is quantitatively defined in Theorem 37, which is a simplified version of Corollary 3.9 in Hanke [1995a]. The bound between the expected risk of the function estimated by MR and the true function is expressed in terms of $|p_k'(0)|$. The theorem uses two technical lemmas proved in Appendix C.3 that bound the distance between an iterate α_k^δ and the true coefficients α^*. Theorem 37 is proved in Appendix C.3. Observe that the bound consists on a decreasing term and an increasing term.

Theorem 37 *Let $R(f)$ be the expected risk $R(f) := \sqrt{\int (f(x) - y)^2 dP(x, y)}$, f_k be the functions associated with the iterates of MR and t the true underlying function. Then*

$$R(f_k) - R(t) \leqslant \frac{a}{|p_k'(0)|} + b|p_k'(0)| + c,$$

where $a = 2\|T^*\|\|K^\dagger\|\|\alpha^*\|, b = \|T^*\|R_{\text{emp}}(t)$ and $c = 2\|K^\dagger\|R_{\text{emp}}(t) + \|\alpha^*\| + \|t - t^*\|$.

The bound consists on a decreasing term $|p_k'(0)|^{-1}$ and an increasing term $|p_k'(0)|$. This implies that f_k initially approaches t then diverges from it. The transition point is dependent on and the empirical risk of the underlying function $R_{\text{emp}}(t)$ and the values of the constants.

5.5 Conclusion

We analysed the idea of regularization by early stopping for conjugate gradient algorithms. We used the idea of a filter function to control the spectrum of the inverse problem, and gave some examples of the filters associated with conjugate gradient methods applied to machine learning datasets from the UCI database. We observe similar behaviour of the filter functions as observed in PLS estimation. We interpreted the idea of semi-convergence of the Krylov subspace iterations of MR in terms of machine learning, and showed that early stopping is essential for proper regularization. In the next chapter, we shall show some empirical evidence that regularization by early stopping can be used for machine learning.

Chapter 6

Applications of Indefinite Kernels

In this chapter, we derive optimization problems associated with indefinite kernels for principal component analysis (PCA), partial least squares (PLS) regression and Fisher discriminant analysis. In the case of PCA, the optimization can be written as a standard eigenvalue problem. The regression and classification problems can be expressed as a system of linear equations. We solve the resulting optimization problems using conjugate gradient type methods (MR, CGNE, and MR-II).

Building on the theoretical framework in Chapter 4, we provide a more practical viewpoint on the matter of learning with indefinite kernels. The aims of this chapter are:

- To perform principal component analysis with indefinite kernels and prove reconstruction error bounds (Section 6.1).
- To show that optimization can be efficiently achieved by using conjugate gradient methods, as theoretically analysed in Chapter 5. This provides another avenue for solving optimization problems associated with kernel methods (including positive semidefinite ones).
- To provide empirical evidence for the efficacy of conjugate gradient type methods with early stopping for solving machine learning problems such as partial least squares regression (Section 6.2) and kernel Fisher discriminant for binary classification (Section 6.3).

We will use the algorithms in Section 5.2 with early stopping for solving regression and binary classification problems in Section 6.2 and 6.3 respectively. First we investigate principal component analysis, which has a simple generalization in the case of the indefinite kernel.

6.1 Principal Component Analysis

Principal component analysis (PCA) is a well known technique for dimensionality reduction. The kernel version [Schölkopf et al., 1999] performs PCA in the feature space induced by the kernel. PCA forms a lower dimensional approximation from the eigenvectors of the operator. For positive semidefinite operators, PCA chooses the largest eigenvectors corresponding to the largest eigenvalues. We show below that the principal components when the kernel is indefinite are the eigenvalues with the largest magnitude. This is motivated by the fact that we want the corresponding largest eigenvalues in the associated Hilbert space $\overline{\mathcal{K}}$ (Definition 18). The main idea is to use the decomposition of a Kreĭn space to two orthogonal Hilbert spaces.

In the following, we denote by lowercase the kernel function $k(x, x')$ and by uppercase the kernel operator K, defined by

$$(Kf)(x) = \int f(y)k(x, y)dP(y).$$

Recall that a Kreĭn space can be decomposed into two orthogonal Hilbert spaces, that is $\mathcal{K} = \mathcal{H}_+ \ominus \mathcal{H}_-$ (Definition 17) and there is an associated majorizing Hilbert space $\overline{\mathcal{K}} = \mathcal{H}_+ \oplus \mathcal{H}_-$. Let $\mathcal{H}_+$ and $\mathcal{H}_-$ have reproducing kernels k_+ and k_-, then the reproducing kernel of $\mathcal{K}$ is $k = k_+ - k_-$ and the reproducing kernel of $\overline{\mathcal{K}}$ is $\overline{k} = k_+ + k_-$. We use the fundamental decomposition in the following derivations. Furthermore, we choose $\mathcal{H}_+$ to contain the span of the eigenvectors of K corresponding to eigenvalues 0.

We want the lower dimensional approximation to be close to the original function. Therefore we define the principal components to be the eigenvectors with associated eigenvalues of largest magnitude. Using the fundamental decomposition, we apply the results of Theorem 6 and 7 of Zwald et al. [2004] to each one of the Hilbert spaces, and obtain bounds on the error of reconstruction. We show the global result below, and omit the localised result. We define $\overline{K}^2$ to be the integral operator associated with kernel $\overline{k}(x_i, x_j)^2$. Following Zwald et al. [2004, Assumption 1], we assume that there exists $M > 0$ such that $k(X, X) \leqslant M$ almost surely.

Let Π_d be the set of all projections of dimension d, and $\pi \in \Pi_d$. Define the expected and empirical reconstruction error as $R(\pi) = \mathbb{E}_X \|k(X, \cdot) - \pi k(X, \cdot)\|^2_{\overline{\mathcal{K}}}$ and $R_m(\pi) = \frac{1}{m}\sum_{i=1}^m \|k(X_i, \cdot) - \pi k(X_i, \cdot)\|^2_{\overline{\mathcal{K}}}$ respectively. Denote by $\pi^*_d = \text{argmin}_\pi R(\pi)$ and $\hat{\pi_d} = \text{argmin}_\pi R_m(\pi)$. Theorem 38 shows that with high probability, when we choose the low dimensional approximation by using the eigenvectors corresponding to the eigenvalues with largest magnitude, the reconstruction error is small.

Theorem 38 *With probability at least* $1 - 6e^{-\xi}$,

$$R(\pi_d^*) - R(\hat{\pi}_d) \leqslant 4\sqrt{\frac{d}{m}\operatorname{tr}(\overline{K}^2)} + 12M\sqrt{\frac{2\xi}{m}}.$$

Proof Since we have a fundamental decomposition, the spaces $\mathcal{H}_+$ and $\mathcal{H}_-$ are orthogonal. Furthermore, each one is positive semidefinite, and hence have positive eigenvalues only. Let u_i be the eigenvectors of K with associated eigenvalues λ_i, that is $Ku_i = \lambda_i u_i$, and let the λ_i be sorted in non-increasing order of magnitude $|\lambda_1(A)| \geqslant |\lambda_2(A)| \geqslant \ldots$. This implies that $\mathcal{K}_+ = \operatorname{span}\{u_i \text{ such that } \lambda_i \geqslant 0\}$ and $\mathcal{K}_- = \operatorname{span}\{u_i \text{ such that } \lambda_i < 0\}$.

Define $\Lambda_d = \{\lambda_1, \lambda_2, \ldots, \lambda_d\}$ the first d eigenvalues, $\Lambda_+ = \{\lambda_i \in \Lambda_d \text{ such that } \lambda_i \geqslant 0\}$, and $\Lambda_- = \{\lambda_i \in \Lambda_d \text{ such that } \lambda_i < 0\}$. Denote the cardinality of the set by, $d_+ = |\Lambda_+|$ and $d_- = |\Lambda_-|$, and observe that $d = d_+ + d_-$.

For $\mathcal{H}_+$ we can apply Zwald et al. [2004, Theorem 6], and obtain that with probability at least $1 - 3e^{-\xi}$,

$$R(\pi_{d_+}^*) - R(\hat{\pi}_{d_+}) \leqslant 4\sqrt{\frac{d_+}{m}\operatorname{tr}(\overline{K}_+^2)} + 6M\sqrt{\frac{2\xi}{m}}.$$

The same bound applied for the negative part. By the union bound, the probability of the total reconstruction error being large is bounded by the sum of the probabilities that the reconstruction error is large in both $\mathcal{H}_+$ and $\mathcal{H}_-$. Since $\operatorname{tr}(\overline{K}_+^2) + \operatorname{tr}(\overline{K}_-^2) \leqslant \operatorname{tr}(\overline{K}^2)$ the result follows. ■

In the following toy example, data was generated using the cosine function in the interval $[-1, 1]$, with some Gaussian noise added to it. The kernel matrix was then centered and the eigenvectors corresponding to the largest absolute valued eigenvalues were computed. This was done using the MATLAB command

```
eigs(K,p,'LM');
```

where K is the kernel matrix, p is the number of output dimensions, and the switch 'LM' selects the eigenvectors corresponding to the eigenvalues with largest magnitude. The results are shown in Figure 6.1. The two kernels used were the Gaussian kernel

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma}\right),$$

and the Epanechnikov kernel

$$k(x_i, x_j) = \left(1 - \frac{\|x_i - x_j\|^2}{\sigma}\right)^2, \text{ for } \frac{\|x_i - x_j\|^2}{\sigma} \leqslant 1.$$

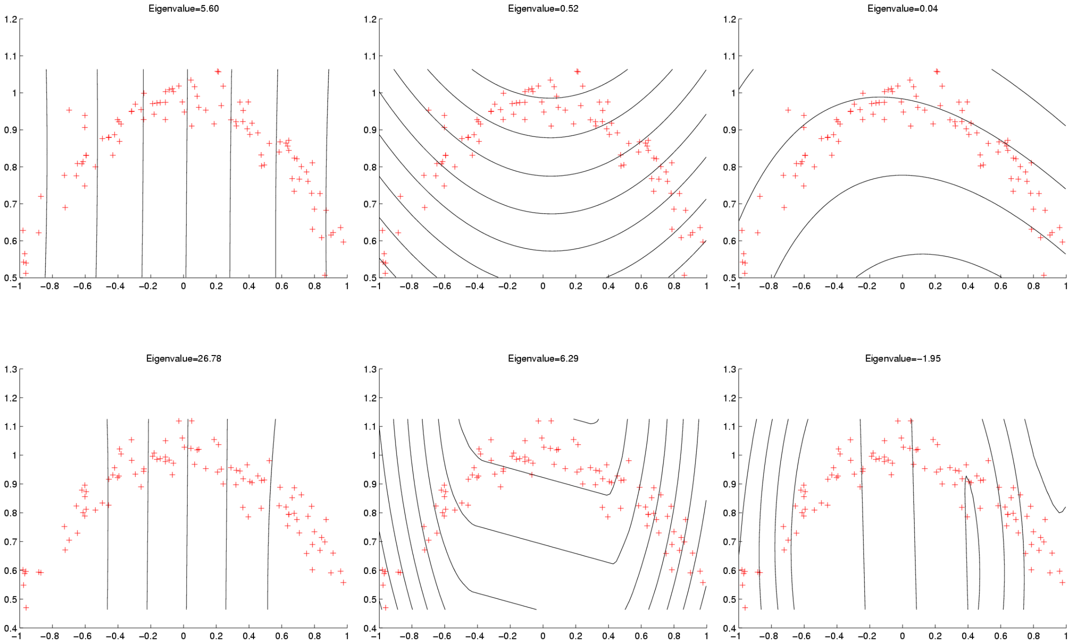


Figure 6.1: Two dimensional toy example. The figures contain lines of constant principal component value (contour lines). We did not draw the eigenvectors as they belong to an infinite dimensional feature space. The first row shows the results using the Gaussian kernel with $\sigma = 0.1$. The second row shows the results using the Epanechnikov kernel with $\sigma = 0.1$.

6.2 Partial Least Squares Regression

We apply the algorithms analysed in Chapter 5, specifically MR, CGNE and MR-II as shown in Figures 5.1 and 5.2. To find an approximation to the interpolation problem such that the solution is in a small subspace of the possible solution space, we choose the subspace to be the Krylov subspace generated by the corresponding conjugate gradient algorithm. This means that we are performing regularization by early stopping of the conjugate gradient algorithm.

Although the problem of regression can be cast as a general eigenvalue problem [Borga et al., 1997], we take the direct approach by using the observation that solving the linear system $K\alpha = y$ using CGNE with early stopping is equivalent to the partial least squares method, which was discussed in Section 5.2.3. We also apply MR and MR-II, hence investigating PLS for different Krylov subspaces.

We used datasets from the UCI Machine Learning repository [Blake and Merz, 1998] for our experiments. The ε-SVR algorithm is taken from SVLAB, with $\varepsilon = 0.1$ and $C = \{10^2, 10^3, 10^4, 10^5, 10^6\}$. The datasets were split as in Meyer et al. [2003], where 10 permutations of 10 fold cross validation were created. The input data was preprocessed to zero mean unit standard deviation. The kernel in the following experiments was the Gaussian kernel and the squared Euclidean distance $k(x, x') = \frac{1}{\sigma}\|x - x'\|^2$, the width was chosen between the 10% and 90% quantile of the squared distance between the input vectors. Observe that the squared Euclidean distance forms a kernel matrix that is not positive semidefinite.

The number of iterations for the iterative methods was tuned using 10 fold cross validation of the training data by computing the cost function on the validation data set. The number of iterations tested was up to 10% of the number of samples in the training data, even though it seems that most of the conjugate gradient type algorithms converge within 20 iterations. We did not explore the effect of different ways of choosing the stopping index. For expositions about the choice of regularization parameters for Krylov subspace methods see Kilmer and O'Leary [2001].

The results (as shown in Table 6.1 and 6.2) indicate that the conjugate gradient type algorithms do not perform as well as ε-SVR or GPR in most of the datasets. However the decrease in performance is within the interquartile range of the best performer. However, the iterative methods are much faster in terms of computations time. In fact, in many instances the computation time is better than using the Matlab matrix inversion with the linear kernel. It is to be noted that the Krylov methods were implemented in Matlab, and performed competitively with the Matlab native code for inversion of a matrix. These gains were due to the fact that only very few iterations were required before the solution was found.

Data	Linear	GPR	ε-SVR	CGNE	MR-II	CGNE*	MR-II*
autompg	10.7±4.5	6.5±2.7	**6.3±3.0**	7.2±3.2	7.2±3.2	8.2±3.8	8.1±3.8
boston	21.7±10.7	9.1±4.8	**8.4±7.5**	13.9±11.1	12.7±8.6	17.4±11.3	16.3±11.6
imports (10^6)	5.5±4.7	4.2±5.0	**3.5±5.4**	8.0±12.2	8.7±11.8	4.4±2.9	4.6±2.9
cpu (10^3)	3.3±3.6	**1.5±2.4**	1.9±2.3	2.8±5.6	2.1±4.5	2.9±5.6	2.5±5.7
servo	1.1±0.4	1.3±0.6	**0.2±0.4**	0.6±0.9	0.6±0.8	0.9±0.5	0.9±0.5

Table 6.1: Regression: mean squared test set errors (median and inter quartile range). The algorithm labelled Linear is found by solving $\alpha = (X^\top X + \lambda\mathbf{I})^{-1}y$. The algorithm labelled GPR is found by solving $\alpha = (K + \lambda\mathbf{I})^{-1}y$ with K generated by the Gaussian kernel. The algorithms labelled CGNE and MR-II were run using the Gaussian kernel. The same algorithms run using Euclidean distance are labelled CGNE* and MR-II*.

By comparing the performance of the Gaussian and the Euclidean distance on the Krylov methods we see that in terms of accuracy, the Gaussian kernel performs better in most cases except for the "imports" dataset. However, the Euclidean distance tends to induce Gram matrices which result in faster convergence.

In summary, the application of Krylov subspace methods to regression gives significant speedup compared to the ε-SVR. However, it was observed that the decrease in computation time increases the error rate by a small amount. This increase in error

Data (items)	Linear	GPR	ε-SVR	CGNE	MR-II	CGNE*	MR-II*
autompg (398)	39	53	553	67(24.6)	57(16.9)	66(19.4)	**34(14.9)**
boston (506)	**72**	97	983	139(34.2)	101(26.9)	153(31.4)	82(**24.8**)
imports (205)	**4**	7	83	5(8.4)	7(7.8)	**4(10.1)**	**4(9.8)**
cpu (209)	8	11	148	11(13.2)	10(12.6)	**5(4.5)**	6(5.2)
servo (167)	6	9	98	6(6.9)	7(**6.4**)	**4(10.1)**	**4(10.0)**

Table 6.2: Regression: mean time for optimization (the mean number of iterations taken by the Krylov subspace algorithms in brackets) The algorithm labelled Linear is found by solving $\alpha = (X^\top X + \lambda\mathbf{I})^{-1}y$. The algorithm labelled GPR is found by solving $\alpha = (K + \lambda\mathbf{I})^{-1}y$ with K generated by the Gaussian kernel. The algorithms labelled CGNE and MR-II were run using the Gaussian kernel. The same algorithms run using Euclidean distance are labelled CGNE* and MR-II*.

was within the interquartile range of the best error.

6.3 Fisher Discriminant

Fisher discriminant analysis (for example [Duda et al., 2000, Section 3.8.2]) is a method for classification. We focus of the binary classification case in this section. The Fisher discriminant maximises a coefficient which is a ratio of between class variances and within class variances. For some domain $\mathcal{X}$, we are given some training data $(X_{train}, Y_{train}) = (x_i, y_i)$ for $i = 1, \ldots, m$, where $x \in \mathcal{X}$ and $y \in \{-1, +1\}$. The kernel Fisher discriminant [Mika et al., 1999] allows us to perform the Fisher discriminant algorithm in feature space. We can derive a similar expression for indefinite kernels. Let $\Phi : \mathcal{X} \to \mathcal{K}$. By the representer theorem, we can express the solution as

$$w = \sum_{i=1}^{m} \alpha_i \Phi(x_i).$$

We define the sample mean for each class in $\mathcal{K}$ to be

$$\mu_\pm = \frac{1}{m_\pm} \sum_{x \in X_\pm} \Phi(x_i) = \frac{1}{m_\pm} K \mathbf{1}_\pm,$$

where K is the gram matrix defined by $K_{ij} = k(x_i, x_j)$ and $\mathbf{1}_+$ and $\mathbf{1}_-$ is a vector with ones in the class $+1$ and -1 respectively and zero elsewhere. We want to maximize the separability of the class centers (the between class variance) and minimize the within class variance. The Fisher discriminant measures this by defining

$$S_B = (\mu_+ - \mu_-)(\mu_+ - \mu_-)^\top,$$

and

$$S_W = \sum_{x \in X_pm} (\Phi(x) - \mu_\pm)(\Phi(x) - \mu_\pm)^\top$$

and maximizing the Rayleigh coefficient [Parlett, 1980]. After some algebraic manipulation (for example see Mika et al. [1999]), the Fisher discriminant can be found by maximizing

$$J(\alpha) = \frac{\alpha^\top M\alpha}{\alpha^\top N\alpha},$$

where

$$N = KK^\top - m_+\mu_+\mu_+^\top - m_-\mu_-\mu_-^\top$$

and

$$M = (\mu_+ - \mu_-)(\mu_+ - \mu_-)^\top.$$

The maximum of Rayleigh coefficient is equivalent to solving the eigenvalue problem $N^{-1}M\alpha = \lambda\alpha$. Since $(\mu_+ - \mu_-)^\top\alpha$ is a scalar, $M\alpha$ is a vector in the direction $\mu_+ - \mu_-$. Since we are interested in the direction of α, we can find the best solution by solving the linear system

$$N\alpha = \mu_+ - \mu_-.$$

Note that we are performing regression on the vector formed by the centers determined by the labels. Therefore, we can use the Krylov subspace techniques of Section 5.2.

We used datasets from the UCI Machine Learning repository for our experiments. The C-SVM results used the implementation in SVLAB, with $C = \{10^2, \ldots, 10^9\}$. The datasets were split as in Meyer et al. [2003], which is 10 permutations of 10 fold cross validation. The input data was preprocessed to zero mean unit standard deviation. The kernel in the following experiments was the Gaussian kernel and the squared Euclidean distance $k(x, x') = \frac{1}{\sigma}\|x - x'\|^2$. The width was chosen between the 25% quartile and the maximum of the squared distance between the input vectors. The early stopping parameter was chosen as in Section 6.2. The results are presented below, with Table 6.3 showing the median and interquartile range of the error rate, and Table 6.4 showing the time taken for training the classifier.

The performance gains in terms of computation time demonstrated in the regression case in the previous section is again apparent for the kernel Fisher discriminant. However, in some of the cases, the magnitude of the speedup is smaller. Interestingly, in several of the datasets tested, the Krylov subspace methods were more accurate than the C-SVM, but still within the inter quartile range. For binary classification, the Euclidean distance had the highest accuracy for the "heart" and "credit" datasets.

Data	*C*-SVM	MR	CGNE	MR-II	MR*	CGNE*	MR-II*
pima	**24.4±3.4**	30.0±3.4	26.7±2.9	26.4±3.1	27.4±2.6	26.4±2.9	26.4±2.8
ionosphere	**5.7±2.1**	35.7±11.4	7.9±3.6	7.1±2.9	25.7±3.6	7.9±2.9	8.6±2.9
wdbc	3.9±3.5	9.2±2.2	**3.5±1.3**	3.9±0.9	10.5±2.6	4.8±1.8	5.3±2.0
heart	17.6±5.0	18.5±5.0	18.5±4.2	17.6±4.2	18.5±4.2	**16.8±3.8**	17.6±3.4
thyroid	**5.8±5.8**	12.8±9.9	**5.8±3.5**	**5.8±2.3**	10.5±3.5	9.3±4.7	9.3±4.7
sonar	15.7±7.2	15.7±4.8	13.3±4.8	**12.0±4.8**	39.8±7.2	24.1±6.0	24.1±6.0
credit	14.4±2.3	15.7±2.7	13.8±2.3	13.8±2.3	21.1±2.9	**13.4±2.7**	13.8±2.7
glass	**8.1±4.7**	8.7±3.5	**8.1±3.5**	**8.1±3.5**	9.3±4.7	**8.1±3.5**	**8.1±3.5**

Table 6.3: Binary Classification: test set errors (median and inter quartile range). The algorithms labelled MR, CGNE and MR-II were run using the Gaussian kernel. The same algorithms run using Euclidean distance are labelled MR*, CGNE* and MR-II*.

Data (items)	*C*-SVM	MR	CGNE	MR-II	MR*	CGNE*	MR-II*
pima (768)	726	546(7.3)	522(34.7)	390(34.7)	372(10.9)	332(12.8)	**269(10.6)**
ionosphere (351)	133	68(4.6)	63(19.9)	36(17.7)	35(**4.9**)	41(17.1)	**34(14.9)**
wdbc (569)	392	232(3.8)	190(26.6)	142(25.3)	**126(4.1)**	167(24.4)	133(27.1)
heart (303)	87	48(8.4)	22(7.9)	21(6.8)	23(**4.7**)	24(10.1)	**19(7.8)**
thyroid (215)	51	17(**2.1**)	9(8.0)	11(8.4)	10(2.6)	9(4.0)	**8(5.1)**
sonar (208)	39	19(8.3)	9(9.4)	10(8.8)	9(**3.6**)	11(9.7)	**8(9.7)**
credit (690)	512	398(14.3)	189(20.8)	185(23.3)	183(**6.1**)	217(20.1)	**157(13.4)**
glass (214)	51	18(**3.4**)	10(8.3)	10(9.2)	9(3.5)	12(6.2)	**8(6.3)**

Table 6.4: Binary Classification: mean time for optimization. For each algorithm, the training time (in miliseconds), was measured using the Matlab command cputime. The mean number of iterations taken by the Krylov subspace algorithms is shown in brackets. The algorithms labelled MR, CGNE and MR-II were run using the Gaussian kernel. The same algorithms run using Euclidean distance are labelled MR*, CGNE* and MR-II*.

6.4 Conclusion

We have shown that several machine learning problems can be solved when using indefinite kernels. Specifically, we have observed that the corresponding KPCA problem can be solved by choosing the principal eigenvectors to be the ones with the eigenvalues of largest magnitude. We have demonstrated that we can solve partial least squares regression and kernel Fisher discriminant when the kernel is indefinite. The resulting optimization problem was solved using Krylov subspace methods.

Using Krylov subspace methods we have given some empirical evidence that partial least squares regression and kernel Fisher discriminant performs comparably with traditional methods in terms of accuracy. In addition, Krylov subspace methods are much faster than the interior point methods used in SVMs. In fact, these gains are also true for the positive semidefinite case. Hence we can widen the types of kernels we use and still optimize the resulting machine learning problems efficiently.

Chapter 7

Conclusion

"Go on, get out - last words are for fools who haven't said enough."

Karl Marx to his housekeeper, who urged him to tell her his last words so she could write them down for posterity, 1883.

This thesis provides a framework for learning the best kernel between objects, a generalization of positive semidefinite kernels and a different approach to regularization. For each extension, we presented some motivating examples, reviewed the relevant results from functional analysis, derived corresponding optimization problems for machine learning and provided experimental results.

In the problem of learning the kernel, the proposed solution is a kernel on the space of kernels itself, called the hyperkernel. This allows us to perform regularization by controlling the norm of the kernel in the Hyper-RKHS. The semidefinite programs corresponding to several machine learning problems were derived. In the experiments with data from the UCI repository, the same parameter settings were used for binary classification, regression and novelty detection. This increased automation still led to results competitive with the state of the art.

The representer theorem for the Hyper-RKHS tells us that the optimum kernel is a linear combination of hyperkernels. However, without further restriction, this may lead to indefinite kernels. Several other motivations for indefinite kernels were also presented, and the functional framework of the RKKS was detailed. We demonstrated that indefinite kernels still gives rise to the representer theorem and generalization error bounds. Since the standard Tikhonov regularization may not be successful in this case, we chose to perform regularization by early stopping. The theoretical properties of this were analyzed via the idea of a filter function on the spectrum of the kernel and an analysis of the rate of convergence of the Minimal Residual algorithm. The idea of semi-convergence is not limited to Krylov subspace algorithms, but can be applied to many iterative algorithms.

The theoretical analysis of Krylov subspace algorithms motivated us to use them for machine learning applications. Although well studied in other fields, Krylov subspace algorithms have not been applied to machine learning yet. Coupled with regularization by early stopping, these algorithms provide a novel way for optimization in machine learning.

In conclusion, the results presented here broaden our understanding of machine learning with kernels: how we regularize them and how we optimize the resulting problem. First, it shows that learning the kernel can be treated in the same framework. Second, it shows that indefinite kernels can be successfully used in machine learning. Third, it proposes regularization by early stopping as an alternative paradigm for machine learning.

Appendix A

Derivations and Proofs for Hyperkernels

A.1 Proof of Proposition 14

We will make use of a theorem due to Albert [1969] which is a generalization of the Schur complement lemma for positive semidefinite matrices.

Theorem 39 (Generalized Schur Complement) *Let* $X = \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix}$, *where* A *and* C *are symmetric. Then*

$$X \succeq 0 \text{ if and only if } A \succeq 0, AA^\dagger B = B \text{ and } C - B^\top A^\dagger B \succeq 0 \tag{A.1}$$

where $A^\dagger$ *is the Moore-Penrose inverse of* A*.*

We prove the proposition that the solution of the quadratic minimax problem (3.4) is obtained by minimizing the SDP (3.5).

Proof Rewrite the terms of the objective function in (3.4) dependent on x in terms of their Wolfe dual. The corresponding Lagrange function is

$$L(x, \xi, \gamma) = -\frac{1}{2} x^\top H(\xi) x - c(\xi)^\top x + \gamma^\top (Ax + a), \tag{A.2}$$

where $\gamma \in \mathbb{R}^M$ is a vector of Lagrange multipliers with $\gamma \geqslant 0$. By differentiating $L(x, \xi, \gamma)$ with respect to x and setting the result to zero, one obtains that (A.2) is maximized with respect to x for $x = H(\xi)^\dagger (A^\top \gamma - c(\xi))$ and subsequently we obtain the dual

$$D(\xi, \gamma) = \frac{1}{2} (A^\top \gamma - c(\xi))^\top H(\xi)^\dagger (A^\top \gamma - c(\xi)) + \gamma^\top a. \tag{A.3}$$

Note that $H(\xi)^\dagger H(\xi) H(\xi)^\dagger = H(\xi)^\dagger$. For equality constraints in (3.4), such as $Bx + b = 0$, we get correspondingly free dual variables.

The dual optimization problem is given by inserting (A.3) into (3.4)

$$\begin{array}{ll} \min\limits_{\xi,\gamma} & \frac{1}{2}(A^\top \gamma - c(\xi))^\top H(\xi)^\dagger (A^\top \gamma - c(\xi)) + \gamma^\top a + d(\xi) \\ \text{subject to} & H(\xi) \succeq 0, G(\xi) \succeq 0, \gamma \geqslant 0. \end{array} \tag{A.4}$$

Introducing an auxiliary variable, t, which serves as an upper bound on the quadratic objective term gives an objective function linear in t and γ. Then (A.4) can be written as

$$\begin{array}{ll} \min\limits_{\xi,\gamma} & \frac{1}{2}t + \gamma^\top a + d(\xi) \\ \text{subject to} & t \succeq (A^\top \gamma - c(\xi))^\top H(\xi)^\dagger (A^\top \gamma - c(\xi)), \\ & H(\xi) \succeq 0, G(\xi) \succeq 0, \gamma \geqslant 0. \end{array} \tag{A.5}$$

From the properties of the Moore-Penrose inverse, we get $H(\xi)H(\xi)^\dagger(A^\top\gamma - c(\xi)) = (A^\top\gamma - c(\xi))$. Since $H(\xi) \succeq 0$, by Theorem 39, the quadratic constraint in (A.5) is equivalent to

$$\left[\begin{array}{cc} H(\xi) & (A^\top \gamma - c(\xi)) \\ (A^\top \gamma - c(\xi))^\top & t \end{array}\right] \succeq 0 \tag{A.6}$$

Stacking all the constraints in (A.5) as one linear matrix inequality proves the claim. ■

A.2 Derivation of Hyperkernel Optimization

This section gives the definitions of SVM optimization problems taken from Schölkopf and Smola [2002], and derive the corresponding hyperkernel optimizations based on the dual of the SVM problem.

In each of the derivations, we replace the empirical quality functional $Q_{\text{emp}}(k, X, Y)$ with the appropriate dual form of the regularized risk functional, in the equation

$$\min_{k \in \underline{\mathcal{H}}} Q_{\text{emp}}(k, X, Y) + \frac{\lambda_Q}{2}\|k\|^2_{\underline{\mathcal{H}}} \tag{A.7}$$

In this subsection, we use the same notation as defined in Section 3.1.2, which is reproduced here for reading convenience. For $p, q, r \in \mathbb{R}^n, n \in \mathbb{N}$ let $r = p \circ q$ be defined as element by element multiplication, $r_i = p_i \times q_i$. The pseudo-inverse (or Moore-Penrose inverse) of a matrix K is denoted $K^\dagger$. Let $\vec{K}$ be the m^2 by 1 vector formed by concatenating the columns of an m by m matrix. We define the hyperkernel Gram matrix $\underline{K}$ by putting together m^2 of these vectors, that is we set $\underline{K} = [\vec{K}_{pq}]_{p,q=1}^m$. Other notations include: the kernel matrix $K = \text{reshape}(\underline{K}\beta)$ (reshaping a m^2 by 1 vector, $\underline{K}\beta$, to a m by m matrix), Y =diag(y) (a matrix with y on the diagonal and zero everywhere else), $G(\beta) = YKY$ (the dependence on β is made explicit), $\mathbf{I}$ the identity matrix, $\mathbf{1}$ a vector of ones and $\mathbf{1}_{m\times m}$ a matrix of ones. Let w be the weight vector and b_{offset} the bias term in feature space, that is the hypothesis function in feature space is defined as $g(x) = w^\top \phi(x) + b_{\mathit{offset}}$ where $\phi(\cdot)$ is the feature mapping defined by the kernel function k.

The number of training examples is assumed to be m, that is $X_{\text{train}} = \{x_1, \ldots, x_m\}$ and $Y_{\text{train}} = y = \{y_1, \ldots, y_m\}$. Where appropriate, γ and χ are Lagrange multipliers, while η and ξ are vectors of Lagrange multipliers from the derivation of the Wolfe dual for the SDP, β are the hyperkernel coefficients, t_1 and t_2 are the auxiliary variables. When $\eta \in \mathbb{R}^m$, we define $\eta \geqslant 0$ to mean that each $\eta_i \geqslant 0$ for $i = 1, \ldots, m$.

A.2.1 L_1 SVM (C-parameterization)

Recall the primal problem for the C-SVM,

$$\begin{aligned} \min_{w,\xi} \quad & \frac{1}{2}\|w\|^2 + \frac{C}{m}\sum_{i=1}^m \xi_i \\ \text{subject to} \quad & y_i(\langle x_i, w\rangle + b) \geqslant 1 - \xi_i \\ & \xi_i \geqslant 0 \text{ for all } i = 1, \ldots, m \end{aligned}$$

and its dual form,

$$\begin{aligned} \max_{\alpha\in\mathbb{R}^m} \quad & \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i=1}^m \alpha_i\alpha_j y_i y_j k(x_i, x_j) \\ \text{subject to} \quad & \textstyle\sum_{i=1}^m \alpha_i y_i = 0 \\ & 0 \leqslant \alpha_i \leqslant \tfrac{C}{m} \text{ for all } i = 1, \ldots, m. \end{aligned}$$

Using the C style parameterization and setting the cost function to the L_1 soft margin loss, that is $c(x_i, y_i, f(x_i)) = \max(0, 1 - y_i f(x_i))$ we get the following equation.

$$\begin{aligned} \min_{f\in\mathcal{H}, k\in\underline{\mathcal{H}}} \quad & \tfrac{C}{m}\textstyle\sum_{i=1}^m \xi_i + \tfrac{1}{2}\|f\|_{\mathcal{H}}^2 + \tfrac{\lambda_Q}{2}\|k\|_{\underline{\mathcal{H}}}^2 \\ \text{subject to} \quad & y_i f(x_i) \geqslant 1 - \xi_i \\ & \xi_i \geqslant 0 \end{aligned}$$

By considering the optimization problem dependent on f, we can use the derivation of the dual problem in the standard C-SVM. The following equation expresses this in matrix notation and also replaces $\|k\|_{\underline{\mathcal{H}}}^2 = \beta^\top \underline{K}\beta$ which is possible due to the representer theorem for hyperkernels.

$$\begin{array}{ll} \min\limits_{\beta} \max\limits_{\alpha} & \mathbf{1}^\top \alpha - \frac{1}{2}\alpha^\top G(\beta)\alpha + \frac{\lambda_Q}{2}\beta^\top \underline{K}\beta \\ \text{subject to} & \alpha^\top y = 0 \\ & 0 \leqslant \alpha_i \leqslant \frac{C}{m} \text{ for all } i = 1, \ldots, m \\ & \beta_i \geqslant 0 \end{array} \tag{A.8}$$

This is the quadratic form of Corollary 15 where $x = \alpha$, $\theta = \beta$, $H(\theta) = G(\beta)$, $c(\theta) = -\mathbf{1}$, $\Sigma = C\lambda_Q\underline{K}$, the constraints are $A = \begin{bmatrix} y & -y & \mathbf{I} & -\mathbf{I} \end{bmatrix}^\top$ and $a = \begin{bmatrix} 0 & 0 & \mathbf{0} & \frac{C}{m}\mathbf{1} \end{bmatrix}^\top$. Applying Corollary 15, we obtain the corresponding SDP. To demonstrate how each constraint leads to the Lagrange multipliers, we replace the matrix constraint with three linear constraints and derive the corresponding semidefinite program for the above optimization problem. We rewrite the terms of A.8 dependent on α in terms of their Wolfe dual. The corresponding Lagrange function is given by the following equation.

$$L(\alpha, \gamma, \eta, \xi) = \mathbf{1}^\top \alpha - \frac{1}{2}\alpha^\top G(\beta)\alpha + \eta^\top \alpha - \xi^\top (\alpha - \frac{C}{m}) + \gamma y^\top \alpha \tag{A.9}$$

where the Lagrange multipliers are γ free, $\eta \geqslant 0$ and $\xi \geqslant 0$. Then, differentiating $L(\alpha, \gamma, \eta, \xi)$ with respect to α gives us:

$$\frac{\partial L(\alpha, \gamma, \eta, \xi)}{\partial \alpha} = \mathbf{1} - G(\beta)\alpha + \eta - \xi + \gamma y$$

Setting this to zero shows that A.9 is minimized with respect to α for $\alpha = G(\beta)^{-1}(\gamma y + \mathbf{1} + \eta - \xi)$. For the case when $G(\beta)$ is positive semidefinite, we can replace the inverse with the Moore-Penrose generalized inverse, and the following results still follow. For notational convenience, let $z = \gamma y + \mathbf{1} + \eta - \xi$.

$$D(\gamma, \eta, \xi) = \frac{1}{2} z^\top G(\beta)^{-1} z + \xi^\top \frac{C}{m}$$

The dual optimization problem is given by inserting the previous equation into A.8.

$$\begin{array}{ll} \min\limits_{\beta, \gamma, \eta, \xi} & \frac{1}{2} z^\top G(\beta)^{-1} z + \xi^\top \frac{C}{m} + \frac{\lambda_Q}{2}\beta^\top \underline{K}\beta \\ \text{subject to} & \eta \geqslant 0, \xi \geqslant 0, \beta \geqslant 0 \end{array} \tag{A.10}$$

Introducing auxilliary variables t_1 and t_2 to bound the quadratic terms in z and β respectively, we get the following optimization problem.

$$\begin{array}{ll} \min\limits_{\beta,\gamma,\eta,\xi} & \frac{1}{2}t_1 + \xi^\top \frac{C}{m} + \frac{\lambda_Q}{2}t_2 \\ \text{subject to} & \eta \geqslant 0, \xi \geqslant 0, \beta \geqslant 0 \\ & t_2 \geqslant \beta^\top \underline{K}\beta \\ & t_1 - z^\top G(\beta)^{-1} z \succeq 0 \end{array} \tag{A.11}$$

Using the Schur Complement Lemma and expressing the quadratic constraint as a second order cone constraint, we get the following semidefinite program.

$$\begin{array}{ll} \min\limits_{\beta,\gamma,\eta,\xi} & \frac{1}{2}t_1 + \xi^\top \frac{C}{m} + \frac{\lambda_Q}{2}t_2 \\ \text{subject to} & \eta \geqslant 0, \xi \geqslant 0, \beta \geqslant 0 \\ & \|\underline{K}^{\frac{1}{2}}\beta\|^2 \leqslant t_2 \\ & \begin{bmatrix} G(\beta) & z \\ z^\top & t_1 \end{bmatrix} \succeq 0. \end{array} \tag{A.12}$$

We replace $\|\underline{K}^{\frac{1}{2}}\beta\|^2 \leqslant t_2$ by $\|\underline{K}^{\frac{1}{2}}\beta\| \leqslant t_2$, and add the scale breaking constraint $\mathbf{1}^\top\beta = 1$, we get

$$\begin{array}{ll} \min\limits_{\beta,\gamma,\eta,\xi} & \frac{1}{2}t_1 + \frac{C}{m}\xi^\top \mathbf{1} + \frac{\lambda_Q}{2}t_2 \\ \text{subject to} & \eta \geqslant 0, \xi \geqslant 0, \beta \geqslant 0 \\ & \|\underline{K}^{\frac{1}{2}}\beta\| \leqslant t_2, \mathbf{1}^\top\beta = 1 \\ & \begin{bmatrix} G(\beta) & z \\ z^\top & t_1 \end{bmatrix} \succeq 0. \end{array}$$

The classification function is given by $f = KG(\beta)^{-1}(z \circ y) - \gamma$.

A.2.2 L_1 SVM (ν-parameterization)

Recall the primal problem for the ν-SVM,

$$\begin{array}{ll} \min\limits_{w,\xi,\rho,b} & \frac{1}{2}\|w\|^2 - \nu\rho + \frac{1}{m}\sum_{i=1}^{m}\xi_i \\ \text{subject to} & y_i(\langle x_i, w\rangle + b) \geqslant \rho - \xi_i \\ & \xi_i \geqslant 0 \text{ for all } i = 1, \ldots, m \\ & \rho \geqslant 0 \end{array}$$

and its dual form

$$\begin{array}{ll} \min\limits_{\alpha \in \mathbb{R}^m} & \frac{1}{2}\sum_{i=1}^{m} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\ \text{subject to} & \sum_{i=1}^m \alpha_i y_i = 0 \\ & \sum_{i=1}^m \alpha_i \geqslant \nu \\ & 0 \leqslant \alpha_i \leqslant \frac{1}{m} \text{ for all } i = 1, \dots, m. \end{array}$$

We derive the regularized quality functional and the SDP that can be used to solve it. Setting the cost function to the L_1 soft margin loss, that is $c(x_i, y_i, f(x_i)) = \max(0, 1 - y_i f(x_i))$, we can use the ν parameterization to get the following equation.

$$\begin{array}{ll} \min\limits_{f \in \mathcal{H}, k \in \underline{\mathcal{H}}} & \frac{1}{m}\sum_{i=1}^m \xi_i - \nu\rho + \frac{1}{2}\|f\|_{\mathcal{H}}^2 + \frac{\lambda_Q}{2}\|k\|_{\underline{\mathcal{H}}}^2 \\ \text{subject to} & y_i f(x_i) \geqslant \rho - \xi_i \\ & \xi_i \geqslant 0 \\ & \rho \geqslant 0 \end{array}$$

By considering the optimization problem dependent on f, we can use the derivation of the dual problem in the standard ν-SVM. The following equation expresses this in matrix notation and also replaces $\|k\|_{\underline{\mathcal{H}}}^2 = \beta^\top \underline{K} \beta$ which is possible due to the representer theorem.

$$\begin{array}{ll} \min\limits_{\beta} \max\limits_{\alpha} & -\frac{1}{2}\alpha^\top G(\beta)\alpha + \frac{\lambda_Q}{2}\beta^\top \underline{K}\beta \\ \text{subject to} & y^\top \alpha = 0 \\ & \mathbf{1}^\top \alpha \geqslant \nu \\ & 0 \leqslant \alpha_i \leqslant \frac{1}{m} \text{ for all } i = 1, \dots, m \\ & \beta_i \geqslant 0 \end{array} \tag{A.13}$$

We derive the corresponding semidefinite program for the above optimization problem. We rewrite the terms of A.13 dependent on α in terms of their Wolfe dual. The corresponding Lagrange function is given by the following equation.

$$L(\alpha, \gamma, \chi, \eta, \xi, \chi) = -\frac{1}{2}\alpha^\top G(\beta)\alpha + \eta^\top \alpha - \xi^\top (\alpha - \frac{1}{m}) + \gamma y^\top \alpha + \chi(\mathbf{1}^\top \alpha - \nu) \tag{A.14}$$

where the Lagrange multipliers are γ free, $\chi \geqslant 0$, $\eta \geqslant 0$ and $\xi \geqslant 0$. Then, differentiating $L(\alpha, \gamma, \chi, \eta, \xi)$ with respect to α gives us:

$$\frac{\partial L(\alpha, \gamma, \chi, \eta, \xi)}{\partial \alpha} = -G(\beta)\alpha + \eta - \xi + \gamma y + \chi \mathbf{1}$$

Setting this to zero shows that A.14 is minimized with respect to α for $\alpha = G(\beta)^{-1}(\gamma y + \chi \mathbf{1} + \eta - \xi)$. For the case when $G(\beta)$ is positive semidefinite, we can replace the inverse with the Moore-Penrose generalized inverse, and the following results still follow. For notational convenience, let $z = \gamma y + \chi \mathbf{1} + \eta - \xi$.

$$D(\gamma, \eta, \xi) = \frac{1}{2} z^\top G(\beta)^{-1} z + \xi^\top \frac{1}{m} - \chi \nu$$

The dual optimization problem is given by inserting the previous equation into A.13.

$$\begin{array}{ll} \min\limits_{\beta,\gamma,\eta,\xi,\chi} & \frac{1}{2} z^\top G(\beta)^{-1} z + \xi^\top \frac{1}{m} - \chi\nu + \frac{\lambda_Q}{2} \beta^\top \underline{K} \beta \\ \text{subject to} & \chi \geqslant 0, \eta \geqslant 0, \xi \geqslant 0, \beta \geqslant 0 \end{array} \tag{A.15}$$

Introducing auxilliary variables t_1 and t_2 to bound the quadratic terms in z and β respectively, we get the following optimization problem.

$$\begin{array}{ll} \min\limits_{\beta,\gamma,\eta,\xi,\chi} & \frac{1}{2} t_1 - \chi\nu + \xi^\top \frac{1}{m} + \frac{\lambda_Q}{2} t_2 \\ \text{subject to} & \chi \geqslant 0, \eta \geqslant 0, \xi \geqslant 0, \beta \geqslant 0 \\ & t_2 \geqslant \beta^\top \underline{K} \beta \\ & t_1 - z^\top G(\beta)^{-1} z \succeq 0 \end{array} \tag{A.16}$$

Using the Schur Complement Lemma and expressing the quadratic constraint as a second order cone constraint, we get the following semidefinite program.

$$\begin{array}{ll} \min\limits_{\beta,\gamma,\eta,\xi,\chi} & \frac{1}{2} t_1 - \chi\nu + \xi^\top \frac{1}{m} + \frac{\lambda_Q}{2} t_2 \\ \text{subject to} & \chi \geqslant 0, \eta \geqslant 0, \xi \geqslant 0, \beta \geqslant 0 \\ & \|\underline{K}^{\frac{1}{2}} \beta\|^2 \leqslant t_2 \\ & \begin{bmatrix} G(\beta) & z \\ z^\top & t_1 \end{bmatrix} \succeq 0 \end{array} \tag{A.17}$$

We replace $\|\underline{K}^{\frac{1}{2}} \beta\|^2 \leqslant t_2$ by $\|\underline{K}^{\frac{1}{2}} \beta\| \leqslant t_2$, and add the scale breaking constraint $\mathbf{1}^\top \beta = 1$, we get the relevant optimization problem. The classification function is given by $f = KG(\beta)^{-1}(z \circ y) - \gamma$.

A.2.3 L_2 SVM (Lagrangian SVM)

Recall the primal problem of the Lagrangian SVM,

$$\begin{aligned} \min_{w,\xi,b} \quad & \frac{\lambda}{2}\left(\|w\|^2 + b^2\right) + \frac{1}{m}\sum_{i=1}^{m} \xi_i^2 \\ \text{subject to} \quad & y_i(\langle \phi(x_i), w\rangle + b) \geqslant 1 - \xi_i \end{aligned}$$

and its associated dual

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^m} \quad & \frac{1}{2}\sum_{i=1}^{m} \alpha_i \alpha_j y_i y_j (K_{ij} + 1 + \lambda m \delta_{ij}) - \sum_{i=1}^{m} \alpha_i \\ \text{subject to} \quad & \alpha_i \geqslant 0 \text{ for all } i = 1, \ldots, m. \end{aligned}$$

Note that $w = \sum_{i=1}^{m} y_i \alpha_i \phi(x_i), b = \sum_{i=1}^{m} \alpha_i, \xi_i = \lambda m \alpha_i$. We derive the regularized quality functional and the SDP that can be used to solve it by setting the cost function to the L_2 soft margin loss, that is

$$c(x_i, y_i, f(x_i)) = \begin{cases} 0 & \text{if } y_i f(x_i) \geqslant 1 \\ (1 - y_i f(x_i))^2 & \text{otherwise} \end{cases}$$

and regularizing the offset constant as well, we obtain the Lagrangian SVM of Mangasarian and Musicant [2001].

$$\begin{aligned} \min_{w,\xi,b,k} \quad & \tfrac{1}{m}\textstyle\sum_{i=1}^{m} \xi_i^2 + \tfrac{\lambda}{2}(\|w\|^2_{\mathcal{H}} + b^2) + \tfrac{\lambda_Q}{2}\|k\|^2_{\underline{\mathcal{H}}} \\ \text{subject to} \quad & y_i(\langle \phi(x_i), w\rangle + b) \geqslant 1 - \xi_i \\ & \xi_i \geqslant 0 \end{aligned}$$

By considering the optimization problem dependent on w and b, we can use the derivation of the dual problem in the standard Lagrangian-SVM. The following equation expresses this in matrix notation and also replaces $\|k\|^2_{\underline{\mathcal{H}}} = \beta^\top \underline{K} \beta$ which is possible due to the representer theorem.

$$\begin{aligned} \min_{\beta} \max_{\alpha} \quad & \mathbf{1}^\top \alpha - \tfrac{1}{2}\alpha^\top H(\beta)\alpha + \tfrac{\lambda_Q}{2}\beta^\top \underline{K}\beta \\ \text{subject to} \quad & \alpha_i \geqslant 0 \text{ for all } i = 1, \ldots, m \\ & \beta \geqslant 0 \end{aligned} \tag{A.18}$$

where $H(\beta) = Y(K + \mathbf{1}_{m \times m} + \lambda m \mathbf{I})Y$. We derive the corresponding semidefinite program for the above optimization problem. We rewrite the terms of A.18 dependent on α

in terms of their Wolfe dual. The corresponding Lagrange function is given by the following equation.

$$L(\alpha,\gamma,\eta) = \mathbf{1}^\top\alpha - \frac{1}{2}\alpha^\top H(\beta)\alpha + \eta^\top\alpha \tag{A.19}$$

where the Lagrange multipliers are $\eta \geqslant 0$ Then, differentiating $L(\alpha,\eta)$ with respect to α gives us:

$$\frac{\partial L(\alpha,\eta)}{\partial\alpha} = \mathbf{1} - H(\beta)\alpha + \eta$$

Setting this to zero shows that A.19 is minimized with respect to α for $\alpha = H(\beta)^{-1}(\mathbf{1}+\eta)$. For the case when $H(\beta)$ is positive semidefinite, we can replace the inverse with the Moore-Penrose generalized inverse, and the following results still follow.

$$D(\eta) = \frac{1}{2}(\eta+\mathbf{1})^\top H(\beta)^{-1}(\eta+\mathbf{1})$$

The dual optimization problem is given by inserting the previous equation into A.18.

$$\begin{array}{ll} \min\limits_{\beta,\eta} & \frac{1}{2}(\eta+\mathbf{1})^\top H(\beta)^{-1}(\eta+\mathbf{1}) + \frac{\lambda_Q}{2}\beta^\top \underline{K}\beta \\ \text{subject to} & \eta \geqslant 0, \beta \geqslant 0 \end{array} \tag{A.20}$$

Introducing auxilliary variables t_1 and t_2 to bound the quadratic terms in w and β respectively, we get the following optimization problem.

$$\begin{array}{ll} \min\limits_{\beta,\eta} & \frac{1}{2}t_1 + \frac{\lambda_Q}{2}t_2 \\ \text{subject to} & \eta \geqslant 0, \beta \geqslant 0 \\ & t_2 \geqslant \beta^\top \underline{K}\beta \\ & t_1 - (\eta+\mathbf{1})^\top H(\beta)^{-1}(\eta+\mathbf{1}) \succeq 0 \end{array} \tag{A.21}$$

Using the Schur Complement Lemma and expressing the quadratic constraint as a second order cone constraint, we get the following semidefinite program.

$$\begin{array}{ll} \min\limits_{\beta,\eta} & \frac{1}{2}t_1 + \frac{\lambda_Q}{2}t_2 \\ \text{subject to} & \eta \geqslant 0, \beta \geqslant 0 \\ & \|\underline{K}^{\frac{1}{2}}\beta\|^2 \leqslant t_2 \\ & \left[\begin{array}{cc} H(\beta) & (\eta+\mathbf{1}) \\ (\eta+\mathbf{1})^\top & t_1 \end{array}\right] \succeq 0 \end{array} \tag{A.22}$$

We replace $\|\underline{K}^{\frac{1}{2}}\beta\|^2 \leqslant t_2$ by $\|\underline{K}^{\frac{1}{2}}\beta\| \leqslant t_2$, and add the scale breaking constraint $\mathbf{1}^\top\beta = 1$, we get the relevant optimization problem. The classification function is given by $f = KH(\beta)^{-1}((\eta+\mathbf{1})\circ y)$.

A.2.4 Singleclass SVM

Recall the novelty detection version of SVM,

$$\begin{aligned}
\min_{w,\xi,\rho} \quad & \frac{1}{2}\|w\|^2 - \rho + \frac{1}{\nu m}\sum_{i=1}^{m}\xi_i \\
\text{subject to} \quad & \langle x_i, w\rangle \geqslant \rho - \xi_i \\
& \xi_i \geqslant 0 \text{ for all } i = 1,\dots,m \\
& \rho \geqslant 0,
\end{aligned}$$

and the dual formulation

$$\begin{aligned}
\min_{\alpha\in\mathbb{R}^m} \quad & \frac{1}{2}\sum_{i=1}^{m}\alpha_i\alpha_j k(x_i,x_j) \\
\text{subject to} \quad & \textstyle\sum_{i=1}^{m}\alpha_i = 1 \\
& 0 \leqslant \alpha_i \leqslant \tfrac{1}{\nu m} \text{ for all } i = 1,\dots,m.
\end{aligned}$$

We derive the regularized quality functional and the SDP that can be used to solve it by setting the cost function to the L_1 soft margin loss, that is $c(x_i, y_i, f(x_i)) = \max(0, 1 - y_i f(x_i))$, and dividing throughout by ν following equation.

$$\begin{aligned}
\min_{f\in\mathcal{H},k\in\underline{\mathcal{H}}} \quad & \tfrac{1}{m\nu}\textstyle\sum_{i=1}^{m}\xi_i - \rho + \tfrac{1}{2}\|f\|^2_{\mathcal{H}} + \tfrac{\lambda_Q}{2\nu}\|k\|^2_{\underline{\mathcal{H}}} \\
\text{subject to} \quad & f(x_i) \geqslant \rho - \xi_i \\
& \xi_i \geqslant 0
\end{aligned}$$

By considering the optimization problem dependent on f, we can use the derivation of the dual problem in the standard single class setting. The following equation expresses this in matrix notation and also replaces $\|k\|^2_{\underline{\mathcal{H}}} = \beta^\top \underline{K}\beta$ which is possible due to the representer theorem.

$$\begin{aligned}
\min_{\beta}\max_{\alpha} \quad & -\tfrac{1}{2}\alpha^\top K\alpha + \tfrac{\lambda_Q}{2\nu}\beta^\top \underline{K}\beta \\
\text{subject to} \quad & \mathbf{1}^\top\alpha = 1 \\
& 0 \leqslant \alpha_i \leqslant \tfrac{1}{\nu m} \text{ for all } i = 1,\dots,m \\
& \beta_i \geqslant 0
\end{aligned} \tag{A.23}$$

We derive the corresponding semidefinite program for the above optimization problem. We rewrite the terms of A.23 dependent on α in terms of their Wolfe dual. The corresponding Lagrange function is given by the following equation.

$$L(\alpha, \gamma, \eta, \xi) = -\frac{1}{2}\alpha^\top K\alpha + \eta^\top\alpha - \xi^\top(\alpha - \frac{1}{\nu m}) + \gamma(\mathbf{1}^\top\alpha - 1) \tag{A.24}$$

where the Lagrange multipliers are γ free, $\eta \geqslant 0$ and $\xi \geqslant 0$. Then, differentiating $L(\alpha, \gamma, \eta, \xi)$ with respect to α gives us:

$$\frac{\partial L(\alpha, \gamma, \eta, \xi)}{\partial \alpha} = -K\alpha + \eta - \xi + \gamma\mathbf{1}$$

Setting this to zero shows that A.24 is minimized with respect to α for $\alpha = K^{-1}(\gamma\mathbf{1} + \eta - \xi)$. For the case when K is positive semidefinite, we can replace the inverse with the Moore-Penrose generalized inverse, and the following results still follow. For notational convenience, let $z = \gamma\mathbf{1} + \eta - \xi$.

$$D(\gamma, \eta, \xi) = \frac{1}{2}z^\top K^{-1}z + \xi^\top\frac{1}{\nu m} - \gamma$$

The dual optimization problem is given by inserting the previous equation into A.23.

$$\begin{array}{ll} \min\limits_{\beta,\gamma,\eta,\xi} & \frac{1}{2}z^\top K^{-1}z + \xi^\top\frac{1}{\nu m} - \gamma + \frac{\lambda_Q}{2\nu}\beta^\top\underline{K}\beta \\ \text{subject to} & \eta \geqslant 0, \xi \geqslant 0, \beta \geqslant 0 \end{array} \tag{A.25}$$

Introducing auxilliary variables t_1 and t_2 to bound the quadratic terms in z and β respectively, we get the following optimization problem.

$$\begin{array}{ll} \min\limits_{\beta,\gamma,\eta,\xi} & \frac{1}{2}t_1 + \xi^\top\frac{1}{\nu m} - \gamma + \frac{\lambda_Q}{2\nu}t_2 \\ \text{subject to} & \eta \geqslant 0, \xi \geqslant 0, \beta \geqslant 0 \\ & t_2 \geqslant \beta^\top\underline{K}\beta \\ & t_1 - z^\top K^{-1}z \succeq 0 \end{array} \tag{A.26}$$

Using the Schur Complement Lemma and expressing the quadratic constraint as a second order cone constraint, we get the following semidefinite program.

$$\begin{array}{ll} \min\limits_{\beta,\gamma,\eta,\xi} & \frac{1}{2}t_1 + \xi^\top\frac{1}{\nu m} - \gamma + \frac{\lambda_Q}{2\nu}t_2 \\ \text{subject to} & \eta \geqslant 0, \xi \geqslant 0, \beta \geqslant 0 \\ & \|\underline{K}^{\frac{1}{2}}\beta\|^2 \leqslant t_2 \\ & \begin{bmatrix} K & z \\ z^\top & t_1 \end{bmatrix} \succeq 0 \end{array} \tag{A.27}$$

The detection function is given by $f = \eta - \xi$.

A.2.5 ν-Regression

Recall the ν-regression primal problem,

$$\begin{array}{cc}\underset{w,\xi,\xi^*,\varepsilon,b}{\min} & \frac{1}{2}\|w\|^2 + C\left(\nu\varepsilon + \frac{1}{m}\sum_{i=1}^{m}(\xi_i + \xi_i^*)\right) \\ \text{subject to} & ((\langle x_i, w\rangle + b) - y_i \leqslant \varepsilon - \xi_i \\ & y_i - ((\langle x_i, w\rangle + b) \leqslant \varepsilon - \xi_i^* \\ & \varepsilon \geqslant 0 \text{ for all } i = 1, \dots, m \\ & \xi_I, \xi_i^* \geqslant 0\end{array}$$

and the associated dual problem

$$\begin{array}{cc}\underset{\alpha,\alpha^*\in\mathbb{R}^m}{\max} & \sum_{i=1}^{m}(\alpha_i^* - \alpha_i)y_i - \frac{1}{2}\sum_{i=1}^{m}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)k(x_i, x_j) \\ \text{subject to} & \sum_{i=1}^{m}(\alpha_i - \alpha_i^*) = 0 \\ & \sum_{i=1}^{m}(\alpha_i^* + \alpha_i) \leqslant C\nu \\ & 0 \leqslant \alpha_i, \alpha_i^* \leqslant \frac{C}{m} \text{ for all } i = 1, \dots, m.\end{array}$$

We derive the regularized quality functional and the SDP that can be used to solve it by setting the cost function to the ε-insensitive loss that is $c(x_i, y_i, f(x_i)) = \max(0, |y_i - f(x_i)| - \varepsilon)$, we can use the ν parameterization to get the following equation.

$$\begin{array}{cc}\underset{f\in\mathcal{H},k\in\underline{\mathcal{H}}}{\min} & \frac{1}{\lambda}\left(\nu\varepsilon + \frac{1}{m}\sum_{i=1}^{m}(\xi_i + \xi_i^*)\right) + \frac{1}{2}\|f\|_{\mathcal{H}}^2 + \frac{\lambda_Q}{2\lambda}\|k\|_{\underline{\mathcal{H}}}^2 \\ \text{subject to} & f(x_i) - y_i \geqslant \varepsilon - \xi_i \\ & y_i - f(x_i) \geqslant \varepsilon + \xi_i^* \\ & \xi_i^{(*)} \geqslant 0 \\ & \varepsilon \geqslant 0\end{array}$$

By considering the optimization problem dependent on f, we can use the derivation of the dual problem in the standard ν-SVR. The following equation expresses this in matrix notation and also replaces $\|k\|_{\underline{\mathcal{H}}}^2 = \beta^\top \underline{K}\beta$ which is possible due to the representer theorem. Define $F(\beta) = \begin{bmatrix} K & -K \\ -K & K \end{bmatrix}$ and $\hat{\alpha} = \begin{bmatrix} \alpha \\ \alpha^* \end{bmatrix}$.

$$\begin{array}{ll} \min\limits_{\beta} \max\limits_{\alpha} & \begin{bmatrix} -y \\ y \end{bmatrix}^\top \hat{\alpha} - \tfrac{1}{2}\hat{\alpha}^\top F(\beta)\hat{\alpha} + \tfrac{\lambda_Q}{2\lambda}\beta^\top \underline{K}\beta \\ \text{subject to} & \begin{bmatrix} \mathbf{1} \\ -\mathbf{1} \end{bmatrix}^\top \hat{\alpha} = 0 \\ & \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}^\top \hat{\alpha} \leqslant \tfrac{\nu}{\lambda} \\ & 0 \leqslant \hat{\alpha} \leqslant \tfrac{1}{m\lambda} \\ \text{for all } i = 1, \dots, 2m & \\ & \beta \geqslant 0 \end{array} \tag{A.28}$$

We derive the corresponding semidefinite program for the above optimization problem. We rewrite the terms of A.28 dependent on α in terms of their Wolfe dual. The corresponding Lagrange function is given by the following equation.

$$\begin{aligned} L(\hat{\alpha}, \gamma, \chi, \eta, \xi) = \; & \begin{bmatrix} -y \\ y \end{bmatrix}^\top \hat{\alpha} - \tfrac{1}{2}\hat{\alpha}^\top F(\beta)\hat{\alpha} - \gamma \begin{bmatrix} \mathbf{1} \\ -\mathbf{1} \end{bmatrix}^\top \hat{\alpha}\eta^\top\hat{\alpha} \\ & -\xi^\top(\hat{\alpha} - \tfrac{1}{m\lambda}) - \chi\left(\begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}^\top \hat{\alpha} - \tfrac{\nu}{\lambda}\right) \end{aligned} \tag{A.29}$$

where the Lagrange multipliers are γ free, $\chi \geqslant 0$, $\eta \geqslant 0$ and $\xi \geqslant 0$. Then, differentiating $L(\hat{\alpha}, \gamma, \chi, \eta, \xi)$ with respect to α gives us:

$$\frac{\partial L(\hat{\alpha}, \gamma, \chi, \eta, \xi)}{\partial \hat{\alpha}} = \begin{bmatrix} -y \\ y \end{bmatrix} - F(\beta)\hat{\alpha} - \gamma \begin{bmatrix} \mathbf{1} \\ -\mathbf{1} \end{bmatrix} + \eta - \xi + \chi\mathbf{1}$$

Setting this to zero shows that A.29 is minimized with respect to $\hat{\alpha}$ for

$$\hat{\alpha} = F(\beta)^{-1}\left(\begin{bmatrix} -y \\ y \end{bmatrix} - \gamma \begin{bmatrix} \mathbf{1} \\ -\mathbf{1} \end{bmatrix} + \eta - \xi + \chi\mathbf{1}\right)$$

For the case when $G(\beta)$ is positive semidefinite, we can replace the inverse with the Moore-Penrose generalized inverse, and the following results still follow. For notational convenience, let $z = \begin{bmatrix} -y \\ y \end{bmatrix} - \gamma \begin{bmatrix} \mathbf{1} \\ -\mathbf{1} \end{bmatrix} + \eta - \xi + \chi\mathbf{1}$.

$$D(\gamma, \eta, \xi, \chi) = \frac{1}{2} z^\top G(\beta)^{-1} z + \xi^\top \frac{1}{m\lambda} + \frac{\chi\nu}{\lambda}$$

The dual optimization problem is given by inserting the previous equation into A.28.

$$\begin{array}{ll} \min\limits_{\beta,\gamma,\eta,\xi,\chi} & \frac{1}{2} z^\top F(\beta)^{-1} z + \xi^\top \frac{1}{m\lambda} - \frac{\chi\nu}{\lambda} + \frac{\lambda_Q}{2\lambda} \beta^\top \underline{K} \beta \\ \text{subject to} & \chi \geqslant 0, \eta \geqslant 0, \xi \geqslant 0, \beta \geqslant 0 \end{array} \tag{A.30}$$

Introducing auxilliary variables t_1 and t_2 to bound the quadratic terms in z and β respectively, we get the following optimization problem.

$$\begin{array}{ll} \min\limits_{\beta,\gamma,\eta,\xi,\chi} & \frac{1}{2} t_1 + \frac{\chi\nu}{\lambda} + \xi^\top \frac{1}{m\lambda} + \frac{\lambda_Q}{2\lambda} t_2 \\ \text{subject to} & \chi \geqslant 0, \eta \geqslant 0, \xi \geqslant 0, \beta \geqslant 0 \\ & t_2 \geqslant \beta^\top \underline{K} \beta \\ & t_1 - z^\top F(\beta)^{-1} z \succeq 0 \end{array} \tag{A.31}$$

Using the Schur Complement Lemma and expressing the quadratic constraint as a second order cone constraint, we get the following semidefinite program.

$$\begin{array}{ll} \min\limits_{\beta,\gamma,\eta,\xi,\chi} & \frac{1}{2} t_1 + \frac{\chi\nu}{\lambda} + \xi^\top \frac{1}{m\lambda} + \frac{\lambda_Q}{2\lambda} t_2 \\ \text{subject to} & \chi \geqslant 0, \eta \geqslant 0, \xi \geqslant 0, \beta \geqslant 0 \\ & \|\underline{K}^{\frac{1}{2}} \beta\|^2 \leqslant t_2 \\ & \begin{bmatrix} F(\beta) & z \\ z^\top & t_1 \end{bmatrix} \succeq 0 \end{array} \tag{A.32}$$

The classification function is given by $f = \begin{bmatrix} -K & K \end{bmatrix} F(\beta)^{-1} z - \gamma$.

Appendix B

Proof of Generalization Bounds

Proof (of Proposition 28).

We first compute an expression which upper bounds the Rademacher average.

$$
\begin{aligned}
\mathbb{E}_\varepsilon \sup_{f \in \mathcal{B}_\mathcal{K}} & \left(\sum_{i=1}^{n} \varepsilon_i f(x_i) \right)^2 \\
= \quad & \mathbb{E}_\varepsilon \sup_{f \in \mathcal{B}_\mathcal{K}} \left(\sum_{i=1}^{n} \varepsilon_i \langle f(.), k(x_i, .) \rangle_\mathcal{K} \right)^2 && \text{(B.1)} \\
= \quad & \mathbb{E}_\varepsilon \sup_{f \in \mathcal{B}_\mathcal{K}} \left(\langle f(.), \sum_{i=1}^{n} \varepsilon_i k(x_i, .) \rangle_\mathcal{K} \right)^2 && \text{(B.2)} \\
= \quad & \mathbb{E}_\varepsilon \left\| \sum_{i=1}^{n} \varepsilon_i \overline{k}(x_i, .) \right\|_{\overline{\mathcal{K}}}^2 && \text{(B.3)} \\
= \quad & \sum_{i=1}^{n} \left\| \overline{k}(x_i, .) \right\|_{\overline{\mathcal{K}}}^2 && \text{(B.4)} \\
= \quad & \sum_{i=1}^{n} \overline{k}(x_i, x_i) = Tr(\overline{K}) && \text{(B.5)}
\end{aligned}
$$

Equation (B.2) is obtained first by using the reproducing property and then because of the linearity of the inner product. Equation (B.3) is due to the Hilbertian topology of underlying Kreĭn spaces (Lemma 27). Equation (B.4) is obtained thanks to a nice property of the Rademacher average [Mendelson, 2003]. The inner product in (B.1) is the norm in the associated Hilbert space. We obtain (B.2) since Rademacher random variables are independent.

By Mendelson [2003, Theorem 15] we know that,

$$
\mathbb{E}_\varepsilon \sup_{f \in \mathcal{B}_\mathcal{K}} \left| \sum_{i=1}^{n} \varepsilon_i f(x_i) \right| \leq \left(\mathbb{E}_\varepsilon \sup_{f \in \mathcal{B}_\mathcal{K}} \left(\sum_{i=1}^{n} \varepsilon_i f(x_i) \right)^2 \right)^{\frac{1}{2}} .
$$

Then using Jensen's inequality, and using the fact that X_i are i.i.d. we have,

$$\begin{aligned}
R_n(\mathcal{B}_{\mathcal{K}}) &\leq \frac{1}{n}\,\mathbb{E}_\mu\big[\mathrm{tr}\,\big(\overline{K}\big)^{\frac{1}{2}}\big] \\
&\leq \frac{1}{n}\,\big[\mathbb{E}_\mu \mathrm{tr}\,\big(\overline{K}\big)\big]^{\frac{1}{2}} \\
&= \frac{1}{\sqrt{n}}\,\big[\mathbb{E}_\mu\,\big(\overline{k}(X,X)\big)\big]^{\frac{1}{2}} \\
&= \frac{1}{\sqrt{n}}\,\big[\int_{\mathcal{X}} \overline{k}(x,x)d\mu(x)\big]^{\frac{1}{2}},
\end{aligned}$$

■

Appendix C

Details of proof of semi-convergence

This section reproduces the proofs of Hanke [1995a, Lemma 3.7 Lemma 3.8].

C.1 Eigenvalue decomposition

Let λ_i, u_i be the eigen system associated with regular matrix K associated with a linear operator from $\mathbb{R}^m$ to $\mathbb{R}^m$ (such that $Ku_i = \lambda_i u_i$). Let p be a polynomial and e a vector in $\mathbb{R}^m$.

$$\begin{aligned}
\|p(K)e\|^2 &= \left\| p(K) \sum_{i=1}^{m} \langle e, u_i \rangle u_i \right\|^2 \\
&= \left\| \sum_{i=1}^{m} \langle e, u_i \rangle p(K) u_i \right\|^2 \\
&= \left\| \sum_{i=1}^{m} \langle e, u_i \rangle p(\lambda_i) u_i \right\|^2 \\
&\leq \sup_{i=1,\ldots,m} p(\lambda_i)^2 \left\| \sum_{i=1}^{m} \langle e, u_i \rangle u_i \right\|^2 \\
&\leq \sup_{i=1,\ldots,m} p(\lambda_i)^2 \sum_{i=1}^{m} \langle e, u_i \rangle^2 \qquad \leq p(\lambda^*)^2 \|e\|^2,
\end{aligned}$$

where λ^* is the eigenvalue that maximizes $\sup_{i=1,\ldots,m} p(\lambda_i)$.

C.2 Lemmas for Bounding the Residual

Let $\theta_{i,k}$ be the ith root of the residual polynomial $p_k(\lambda)$ of order k. An important root for the proof is $\theta_{1,k}$, the root closest to zero (see Figure C.1). Recall that we can express a polynomial as a product of its roots, that is

$$p_k(\lambda) = \prod_{i=1}^{k}\left(1 - \frac{\lambda}{\theta_{i,k}}\right) \quad \text{and hence } p_k'(0) = -\sum_{i=1}^{k}\frac{1}{\theta_{i,k}}.$$

Note that $p_k'(0) < 0$ and $|p_k'(0)| \geqslant k$ and recall that λ_i are the eigenvalues of K the kernel matrix sorted in decreasing order. For two polynomials, we define their orthogonality via the inner product

$$\langle \phi, \psi \rangle_{poly} := \langle \phi(K)y, K\psi(K)y \rangle_{\mathbb{R}^m},$$

where $y \in \mathbb{R}^m$. Define $\|v\|_{(a,b)} := \sqrt{\sum_{i=a}^{b} v_i^2}$ and let

$$\varphi_k(\lambda) := p_k(\lambda)\left(\frac{\theta_{1,k}}{\theta_{1,k} - \lambda}\right)^{\frac{1}{2}}, \text{ for } 0 \leqslant \lambda \leqslant \theta_{1,k}.$$

We prove four technical lemmas which are then used to prove Theorem 36.

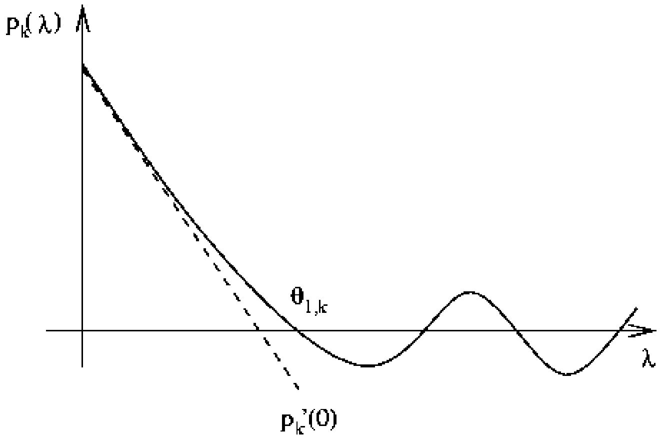


Figure C.1: The parameter of interest is based on the gradient of the residual polynomial, $p_k'(0)$. Note that $p_k(\lambda)$ is convex in $[0, \theta_{1,k}]$.

Lemma 40 *Choose j such that $\lambda_j \geqslant \theta_{1,k} > \lambda_{j+1}$. Then*

$$\sum_{i=1}^{j} p_k^2(\lambda_i)\frac{\lambda_i}{\theta_{1,k} - \lambda_i}(y^\delta)_i^2 \geqslant \sum_{i=j+1}^{m} p_k^2(\lambda_i)(y^\delta)_i^2.$$

Proof Since $p_k(\lambda)$ are residual polynomials, they are orthogonal, and hence $p_k(\lambda)$ and $\frac{p_k(\lambda)}{\lambda-\theta_{1,k}}$ are orthogonal. That is

$$\langle p_k(\lambda)y^\delta, \lambda\frac{p_k(\lambda)}{\lambda-\theta_{1,k}}y^\delta\rangle = 0.$$

This means that

$$\sum_{i=1}^{m} p_k^2(\lambda_i)(y^\delta)_i^2\frac{\lambda_i}{\lambda_i-\theta_{1,k}} = 0.$$

Splitting the sum into two parts

$$\sum_{i=1}^{j} p_k^2(\lambda_i)(y^\delta)_i^2\frac{\lambda_i}{\theta_{1,k}-\lambda_i} = \sum_{i=j+1}^{m} p_k^2(\lambda_i)(y^\delta)_i^2\frac{\lambda_i}{\lambda_i-\theta_{1,k}}.$$

Since $\frac{\lambda_i}{\lambda_i-\theta_{1,k}} \geqslant 1$, the right hand side is greater than $\sum_{i=j+1}^{m} p_k^2(\lambda_i)(y^\delta)_i^2$ and the result follows. ■

Lemma 41 *Choose j such that $\lambda_j \geqslant \theta_{1,k} > \lambda_{j+1}$ and let $v \in \mathbb{R}^m$ be any vector. Then*

$$\|p_k(K)v\| \leqslant \|\varphi_k(K)v\|_{(1,j)}.$$

Proof

$$\begin{aligned}
\|p_k(K)v\| &= \sqrt{\textstyle\sum_{i=1}^{m} p_k^2(\lambda_i)v^2} \\
&= \sqrt{\textstyle\sum_{i=1}^{j} p_k^2(\lambda_i)v^2 + \sum_{i=j+1}^{m} p_k^2(\lambda_i)v^2} && \text{where } j \text{ is as in Lemma 40} \\
&\leqslant \sqrt{\textstyle\sum_{i=1}^{j} p_k^2(\lambda_i)v^2 + p_k^2(\lambda_i)\frac{\lambda_i}{\theta_{1,k}-\lambda_i}v^2} \\
&= \sqrt{\textstyle\sum_{i=1}^{j} p_k^2(\lambda_i)v^2\left(1+\frac{\lambda_i}{\theta_{1,k}-\lambda_i}\right)} \\
&= \sqrt{\textstyle\sum_{i=1}^{j} p_k^2(\lambda_i)v^2\left(\frac{\theta_{1,k}}{\theta_{1,k}-\lambda_i}\right)} \\
&= \sqrt{\textstyle\sum_{i=1}^{j} \varphi_k^2(\lambda_i)v^2} && \text{using the definition of } \varphi_k(\lambda_i) \\
&= \|\varphi_k(K)v\|_{(1,j)}.
\end{aligned}$$

■

Lemma 42

$$\varphi_k^2(\lambda) \leqslant 1 \textit{ for } 0 \leqslant \lambda \leqslant \theta_{1,k}.$$

Proof Observe that $p_k(\lambda) \leqslant 1 - \frac{\lambda}{\theta_{1,k}}$ for $\lambda \in [0, \theta_{1,k}]$. The result follows by some algebraic manipulation of $p_k^2(\lambda)$,

$$\begin{aligned} &p_k^2(\lambda) \leqslant 1 - \tfrac{\lambda}{\theta_{1,k}} \\ &p_k^2(\lambda)\left(\tfrac{\theta_{1,k}}{\theta_{1,k}-\lambda}\right) \leqslant 1. \end{aligned}$$

■

Lemma 43

$$\lambda^2 \varphi_k^2(\lambda) \leqslant 4|p_k'(0)|^{-2} \textit{ for } 0 \leqslant \lambda \leqslant \theta_{1,k}.$$

Proof Let $J(\lambda) = \lambda^2 \varphi_k^2(\lambda)$. The maximum value of $J(\lambda)$ occurs at the point where $J'(\lambda) = 0$. Substituting the definition of $\varphi_k^2(\lambda)$, we get

$$J(\lambda) = \lambda^2 \prod_{i=1}^{k}\left(1 - \frac{\lambda}{\theta_{i,k}}\right)\prod_{i=2}^{k}\left(1 - \frac{\lambda}{\theta_{i,k}}\right).$$

Hence

$$J'(\lambda) = \lambda \prod_{i=1}^{k}\left(1 - \frac{\lambda}{\theta_{i,k}}\right)\prod_{i=2}^{k}\left(1 - \frac{\lambda}{\theta_{i,k}}\right)\left(2 + \frac{\lambda}{\theta_{1,k} - \lambda} - \sum_{i=1}^{k}\frac{2\lambda}{\theta_{i,k} - \lambda}\right).$$

Let λ^* be the point at which the maximum is attained, and hence

$$2 - \sum_{i=1}^{k}\frac{\lambda^*}{\theta_{i,k} - \lambda^*} = -\frac{\lambda^*}{\theta_{i,k} - \lambda^*} + \sum_{i=1}^{k}\frac{\lambda^*}{\theta_{i,k} - \lambda^*} = \sum_{i=2}^{k}\frac{\lambda^*}{\theta_{i,k} - \lambda^*} \geqslant 0.$$

Therefore

$$2 \geqslant \sum_{i=1}^{k}\frac{\lambda^*}{\theta_{i,k} - \lambda^*} \geqslant \sum_{i=1}^{k}\frac{\lambda^*}{\theta_{i,k}} = \lambda^*|p_k'(0)|.$$

The result is obtained by chaining the inequalities together

$$\lambda^2 \varphi_k^2(\lambda) \leqslant \lambda^{*2}\varphi_k^2(\lambda^*) \leqslant \frac{4}{|p_k'(0)|^2}.$$

■

Proof [of Theorem 36]

By the definition of empirical risk and the residual polynomial of MR,

$$\begin{aligned} R_{\text{emp}}(f_k) &= \|y^\delta - K\alpha_k^\delta\| \\ &= \|p_k(K)y^\delta\| && \text{where we assumed } \alpha_0^\delta = 0 \\ &\leqslant \|\varphi_k(K)y^\delta\| && \text{by Lemma 41,} \end{aligned}$$

where $\varphi_k(\lambda) := p_k(\lambda)\left(\frac{\theta_{1,k}}{\theta_{1,k}-\lambda}\right)^{\frac{1}{2}}$, for $0 \leqslant \lambda \leqslant \theta_{1,k}$ and $\theta_{1,k}$ is the first root of the residual polynomial $p_k(\lambda)$. We can now bound the norm of the residual using the norm inequalities

$$\begin{aligned} R_{\text{emp}}(f_k) &\leqslant \|\varphi_k(K)(y^\delta - y)\| + \|\varphi_k(K)y\| && \text{using the triangle inequality} \\ &\leqslant \|y^\delta - y\| + \|\varphi_k(K)K\alpha^*\| && \text{by Lemma 42 and } K\alpha^* = y \\ &\leqslant \|y^\delta - y\| + 2|p_k'(0)|^{-1}\|\alpha^*\| && \text{by Lemma 43.} \end{aligned}$$

The result is obtained by observing that $R_{\text{emp}}(t) = \|y^\delta - K\alpha^*\| = \|y^\delta - y\|$. ■

C.3 Lemmas for Bounding the Error

In the results below recall that $\|v\|_{(a,b)} := \sqrt{\sum_{i=a}^{b} v_i^2}$.

Lemma 44 *We choose j such that $\lambda_j \leqslant |p_k'(0)|^{-1}$, then*

$$\|\alpha_k^\delta - \alpha^*\|_{(1,j)} \leqslant |p_k'(0)|R_{\text{emp}}(t) + \|\alpha^*\|.$$

Proof Recall that q_{k-1} is the iteration polynomial associated with applying MR to the problem $K\alpha = y^\delta$. Define $\hat{\alpha_k} := q_{k-1}(K)y$, and observe that this is not the same as applying MR to the unperturbed data y, since in general, this would result in a different polynomial. Note that since $p_k(\lambda) = 1 - \lambda q_{k-1}(\lambda)$ we can derive $\alpha - \hat{\alpha_k} = p_k(K)\alpha$.

$$\begin{aligned} \|\alpha_k^\delta - \alpha^*\|_{(1,j)} &\leqslant \|\alpha_k^\delta - \hat{\alpha}\|_{(1,j)} + \|\hat{\alpha} - \alpha^*\|_{(1,j)} \\ &\leqslant \|q_{k-1}(K)(y^\delta - y)\|_{(1,j)} + \|p_k(K)\alpha^*\|_{(1,j)} \\ &\leqslant \|q_{k-1}(K)\|\|y^\delta - y\|_{(1,j)} + \|\varphi_k(K)\alpha^*\|_{(1,j)}, \end{aligned}$$

where the last line was obtained by application of Lemma 41. Since p_k is convex in $[0, \lambda_j]$,

$$0 \leqslant q_{k-1}(\lambda) = \frac{1 - p_k(\lambda)}{\lambda} \leqslant |p_k'(0)|.$$

Using Lemma 42, and the bound on q_{k-1}, we obtain the result. ■

Lemma 45 *For the value of j chosen in Lemma 44,*

$$\|\alpha_k^\delta - \alpha^*\|_{(j+1,m)} \leqslant 2\|K^\dagger\| \left(R_{\text{emp}}(t) + \frac{1}{|p_k'(0)|}\|\alpha^*\| \right).$$

Proof

$$\begin{aligned} \|\alpha_k^\delta - \alpha\|_{(j+1,m)} &= \|K^\dagger K\alpha_k^\delta - K^\dagger y\|_{(j+1,m)} \\ &\leqslant \|K^\dagger\|\|K\alpha_k^\delta - y\|_{(j+1,m)} \\ &\leqslant \|K^\dagger\| \left(\|K\alpha_k^\delta - y^\delta\|_{(j+1,m)} + \|y^\delta - y\|_{(j+1,m)} \right). \end{aligned}$$

where the last line was obtained by the triangle inequality. Recall the definition of $R_{\text{emp}}(f_k) = \|y^\delta - K\alpha_k^\delta\|$ and $R_{\text{emp}}(t) = R_{\text{emp}}(t^*) = \|y^\delta - y\|$. By applying Theorem 36, we get the result. ■

Proof [of Theorem 37]

The difference between the real risk on the function associated with iteration k of MR and the true function is given by,

$$R(f_k) - R(t) \leqslant \|f_k - t\|.$$

Define t^* to be the projection of the unknown function t onto the domain of T^*, that is $t^* = T^*\alpha^*$ where α^* is the solution of the unperturbed problem $K\alpha^* = y$. Then by the triangle inequality,

$$\begin{aligned} \|f_k - t\| &\leqslant \|f_k - t^*\| + \|t^* - t\| \\ &= \|T^*\alpha_k^\delta - T^*\alpha\| + \|t^* - t\| \\ &\leqslant \|T^*\|\|\alpha_k^\delta - \alpha\| + \|t^* - t\|. \end{aligned}$$

We choose j such that the jth eigenvalue of K denoted $\lambda_j \leqslant |p_k'(0)|^{-1}$. Observe that

$$\|\alpha_k^\delta - \alpha\| \leqslant \|\alpha_k^\delta - \alpha\|_{(1,j)} + \|\alpha_k^\delta - \alpha\|_{(j+1,m)},$$

where we define $\|v\|_{(a,b)} := \sqrt{\sum_{i=a}^b v_i^2}$. We apply Lemma 44 to bound the first term on the right hand side and Lemma 45 to bound the second term. Hence

$$\|\alpha_k^\delta - \alpha\| \leqslant |p_k'(0)|R_{\text{emp}}(t) + \|\alpha^*\| + 2\|K^\dagger\| \left(R_{\text{emp}}(t) + \frac{1}{|p_k'(0)|}\|\alpha^*\| \right),$$

and the result follows. ■

Bibliography

N.I. Akhiezer and I.M. Glazman. *Theory of Linear Operators in Hilbert Space.* Dover Publications, Inc., 1993.

Arthur Albert. Conditions for positive and nonnegative definiteness in terms of pseudoinverses. *SIAM Journal on Applied Mathematics*, 17(2):434–440, 1969.

Arthur Albert. *Regression and the Moore-Penrose Pseudoinverse*, volume 94 of *Mathematics in Science and Engineering.* Academic Press, Inc., 1972.

D. Alpay. *The Schur algorithm, reproducing kernel spaces and system theory*, volume 5 of *SMF/AMS Texts and Monographs.* Société Mathématique de France, 2001.

D. Alpay, A. Dijksma, J. Rovnyak, and H. S. V. de Snoo. Reproducing kernel pontryagin spaces. In S. Axler, J. McCarthy, and D. Sarason, editors, *Holomorphic Spaces*, volume 33, pages 425–444. Cambridge University Press, 1997.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

Owe Axelsson. *Iterative Solution Methods.* Cambridge University Press, 1994.

T. Ya. Azizov and I. S. Iokhvidov. *Linear Operators in Spaces with an Indefinite Metric.* John Wiley & Sons, 1989. Translated by E. R. Dawson.

F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

P. L. Bartlett and J. Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods—Support Vector Learning*, pages 43–54, Cambridge, MA, 1999. MIT Press.

K. P. Bennett and M. J. Embrechts. An optimization perspective on partial least squares. In J.A.K. Suykens, G. Horvath, S. Basu, C. Micchelli, and J. Vandewalle, editors, *Advances in Learning Theory: Methods, Models and Applications, NATO Science Series III: Computer & Systems Sciences*, volume 190, pages 227–250, 2003.

K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992.

Michele Benzi. Preconditioning techniques for large linear systems: A survey. *Journal of Computational Physics*, 182:418–477, 2002.

Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 1995.

C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998. Available at http://www.ics.uci.edu/∼mlearn/MLRepository.html.

János Bognár. *Indefinite Inner Product Spaces.* Springer Verlag, 1974.

M. Borga, T. Landelius, and H. Knutsson. A unified approach to PCA, PLS, MLR and CCA. Technical Report LiTH-ISY-R-1992, ISY, SE-581 83 Linköping, Sweden, 1997.

Olivier Bousquet and Daniel J.L. Herrmann. On the complexity of learning the kernel matrix. In *Advances in Neural Information Processing Systems 15*, 2002.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

Claude Brezinski. Projection methods for linear systems. *Journal of Computational and Applied Mathematics*, 77:35–51, 1997.

Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

Neil A. Butler and Michael C. Denham. The peculiar shrinkage properties of partial least squares regression. *Journal of the Royal Statistical Society, Series B (Methodological)*, 62(3):585–593, 2000.

D. Calvetti, B. Lewis, and L. Reichel. Smooth or abrupt: a comparison of regularization methods. In F.T. Luk, editor, *Advanced Signal Processing Algorithms, Architectures and Implementations VIII*, pages 286–295, 1998.

D. Calvetti and L. Reichel. Lanczos-based exponential filtering for discrete ill-posed problems. *Numerical Algorithms*, 29(1–3):134–149, 2002.

Daniela Calvetti and Lothar Reichel. Tikhonov regularization of large linear problems. *BIT Numerical Mathematics*, 43:263–283, 2003.

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

Zhe Chen and Simon Haykin. On different facets of regularization theory. *Neural Computation*, 14(12):2791–2846, 2002.

C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

Koby Crammer, Joseph Keshet, and Yoram Singer. Kernel design using boosting. In *Advances in Neural Information Processing Systems 15*, 2002.

Nello Cristianini, Jaz Kandola, Andre Elisseeff, and John Shawe-Taylor. On optimizing kernel alignment. *Journal of Machine Learning Research*, 2003. submitted.

Nello Cristianini, John Shawe-Taylor, André Elisseeff, and Jaz Kandola. On kernel-target alignment. In *Neural Information Processing Systems*, 2001. URL `citeseer.nj.nec.com/480615.html`.

Sijmen de Jong. PLS fits closer than PCR. *Journal of Chemometrics*, 7:551–557, 1993.

Sijmen de Jong. PLS shrinks. *Journal of Chemometrics*, 9:323–326, 1995.

Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer Verlag, 1996.

Michael A. Dritschel and James Rovnyak. Operators on indefinite inner product spaces. In *Lectures on operator theory and its applications*, volume 3 of *Fields Institute Monographs*, pages 141–232, 1996.

K. Duan, S.S. Keerthi, and A.N. Poo. Evaluation of simple performance measures for tuning SVM hyperparameters. *Neural Computation*, 51:41–59, 2003.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification.* John Wiley & Sons, second edition, 2000.

Heinz W. Engl. Regularization methods for the stable solution of inverse problems. In *Lecture Notes for Summer School 'Jacques Louis Lions' Multidisciplinary Methods for Analysis, Optimization and Control of Complex Systems*, 2003. The Lecture Notes will be published in the Mathematics in Industry Series of Sprniger-Verlag.

Heinz W. Engl and Philipp Kügler. Nonlinear inverse problems: Theoretical aspects and some industrial applications. Inverse Problems: Computational Methods and Emerging Applications Tutorials, UCLA, 2003.

Shai Fine and Katya Scheinberg. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2001.

Bernd Fischer, Martin Hanke, and Marlis Hochbruck. A note on conjugate-gradient type methods for indefinite and/or inconsistent linear systems. *Numerical Algorithms*, 11:181–189, 1996.

Harald Frankenberger and Martin Hanke. Kernel polynomials for the solution of indefinite and ill-posed problems. *Numerical Algorithms*, 25:197–212, 2000.

Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996. URL `citeseer.nj.nec.com/freund96experiments.html`.

F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7:219–269, 1995.

Lev Goldfarb. A new approach to pattern recognition. In L. N. Kanal and A. Rosenfeld, editors, *Progress in Pattern Recognition 2*, pages 241–402, 1985. Chapter 9.

Gene H. Golub and Charles F. van Loan. *Matrix Computations*. The John Hopkins University Press, 3rd edition, 1996.

Constantinos Goutis. Partial least squares algorithm yields shrinkage estimators. *The Annals of Statistics*, 24(2):816–824, 1996.

T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and K. Obermayer. Classification on pairwise proximity data. In *Advances in Neural Information Processing Systems 11*, pages 438–444, 1999.

Anne Greenbaum. *Iterative Methods for Solving Linear Systems*. Number 17 in Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics, 1997.

Charles W. Groetsch. *Generalized Inverses of Linear Operators: Representation and Approximation*. Marcel Dekker, Inc., 1977.

Charles W. Groetsch. *The Theory of Tikhonov Regularization for Fredholm equations of the first kind*. Number 105 in Research Note in Mathematics. Pitman Advanced Publishing Program, 1984.

Bernard Haasdonk. Feature space interpretation of SVMs with non positive definite kernels. Unpublished, 2003.

Eldad Haber. *Numerical Strategies for the Solution of Inverse Problems*. PhD thesis, Institute of Applied Mathematics, University of British Columbia, 1997.

M. Hanke and C. W. Groetsch. Nonstationary iterated Tikhonov regularization. *Journal of Optimization Theory and Applications*, 98(1):37–53, 1998.

M. Hanke and P.C. Hansen. Regularization methods for large-scale problems. *Surveys on Mathematics for Industry*, 3:253–315, 1993.

Martin Hanke. *Conjugate Gradient Type Methods for Ill-Posed Problems.* Pitman Research Notes in Mathematics Series. Longman Scientific & Technical, 1995a.

Martin Hanke. The minimal error conjugate gradient method is a regularization method. *Proceedings of the American Mathematical Society*, 123(11):3487–3497, 1995b.

Per Christian Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion.* Society for Industrial and Applied Mathematics, 1998.

Babak Hassibi, Ali H. Sayed, and Thomas Kailath. *Indefinite-Quadratic Estimation and Control: A Unified Approach to H^2 and H^∞ Theories.* Society for Industrial and Applied Mathematics, 1999.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning.* Springer Series in Statistics. Springer Verlag, 2001.

David Haussler. Convolution kernels on discrete structures. Technical Report UCS-CRL-99-10, UC Santa Cruz, 1999. URL `citeseer.nj.nec.com/haussler99convolution.html`.

Simon Haykin. *Neural Networks: A comprehensive foundation.* Prentice Hall, Inc., second edition, 1999.

Inge S. Helland. On the structure of partial least squares regression. *Comminications in Statistics: Simulation and Computation*, 17(2):581–607, 1988.

Inge S. Helland. Some theoretical aspects of partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 58:97–107, 2001.

Ralf Herbrich and Robert C. Williamson. Algorithmic luckiness. *Journal of Machine Learning Research*, 3:175–212, 2002.

Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards.*, 49:409–436, 1952.

A.E. Hoerl and R.W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(3):55–67, 1970.

S. Hyvönen and O. Nevanlinna. Robust bounds for Krylov methods. *BIT*, 40(2): 267–290, 2000.

Misha Kilmer and G. W. Stewart. Iterative regularization and MINRES. *SIAM Journal on Matrix Analysis and Applications*, 21(2):613–628, 1999.

Misha E. Kilmer and Dianne P. O'Leary. Choosing regularization parameters in iterative methods for ill-posed problems. *SIAM Journal on Matrix Analysis and Applications*, 22(4):1204–1221, 2001.

G. S. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82–95, 1971.

Gert Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semi-definite programming. In *Proceedings of the Nineteenth International Conference of Machine Learning*, pages 323–330, 2002.

Gert Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semi-definite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards.*, 45:255–282, 1950.

Cornelius Lanczos. Solutions of systems of linear equations by minimized iterations. *Journal of Research of the National Bureau of Standards.*, 49:33–53, 1952.

Steffen L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

H.-T. Lin and C.-J. Lin. A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. March, 2003.

Ole C. Lingjærde and Nils Christophersen. Shrinkage structure of partial least squares. *Scandinavian Journal of Statistics*, 27(3):459–473, 2000.

J. Löfberg. YALMIP, yet another LMI parser, 2002. http://www.control.isy.liu.se/˜johanl/yalmip.html.

A. Luntz and V. Brailovsky. On estimation of characters obtained in statistical procedure of recognition. *Technicheskaya Kibernetica*, 3, 1969. (in Russian).

D. J. C. MacKay. Bayesian non-linear modelling for the energy prediction competition. *American Society of Heating, Refrigerating and Air-Conditioning Engineers Transcations*, 4:448–472, 1994.

O. L. Mangasarian and D. R. Musicant. Lagrangian support vector machines. *Journal of Machine Learning Research*, 1:161–177, 2001.

Xavier Mary. *Hilbertian subspaces, subdualities and applications.* PhD thesis, Institut National des Sciences Appliquees Rouen, 2003.

Shahar Mendelson. A few notes on statistical learning theory. In S. Mendelson and A. J. Smola, editors, *Advanced Lectures in Machine Learning*, LCNS2600, pages 1–40. Springer Verlag, 2003.

David Meyer, Friedrich Leisch, and Kurt Hornik. The Support Vector Machine under test. *Neurocomputing*, 55(1–2):169–186, 2003.

Charles Micchelli and Massimiliano Pontil. A function representation for learning in Banach spaces. In *Proceedings of the Conference on Learning Theory, 2004*, pages 255–269, 2004.

S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48, 1999.

V.A. Morozov. *Methods for solving incorrectly posed problems.* Springer Verlag, 1984.

Frank Natterer. On the order of regularization methods. *Improperly Posed Problems and Their Numerical Treatment*, pages 189–203, 1982.

R. Neal. *Bayesian Learning in Neural Networks.* Springer Verlag, 1996.

Arkadi S. Nemirovskii. The regularization properties of the adjoint gradient method in ill-posed problems. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 26(2):7–16, 1986.

Cheng Soon Ong, Xavier Mary, Stéphane Canu, and Alexander J. Smola. Learning with non-positive kernels. In *Proceedings of the 16th International Conference on Machine Learning*, pages 639–646, 2004.

Cheng Soon Ong and Alexander J. Smola. Machine learning with hyperkernels. In *International Conference of Machine Learning*, pages 568–575, 2003.

Cheng Soon Ong, Alexander J. Smola, and Robert C. Williamson. Hyperkernels. In *Advances in Neural Information Processing Systems 15*, pages 495–502, 2003.

M. Opper and O. Winther. Gaussian processes and SVM: Mean field and leave-one-out. In A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 311–326, Cambridge, MA, 2000. MIT Press.

Beresford N. Parlett. *The Symmetric Eigenvalue Problem.* Prentice Hall series in Computational Mathematics. Prentice Hall, Inc., 1980.

Elżbieta Pekalska, Pavel Paclík, and Robert P.W. Duin. A generalized kernel approach to dissimilarity-based classification. *Journal of Machine Learning Research*, 2:175–211, 2001.

Aloke Phatak and Frank de Hoog. Exploiting the connection between PLS, Lanczos methods and conjugate gradients: alternative proofs of some properties of PLS. *Journal of Chemometrics*, 16:361–367, 2002.

G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, March 2001.

Gunnar Rätsch, Alexander J. Smola, and Sebastian Mika. Adapting codes and embeddings for polychotomies. In *Advances in Neural Information Processing Systems 15*, 2002.

Michael Reed and Barry Simon. *Methods of Modern Mathematical Physics*, volume 1 of *Functional Analysis.* Academic Press, Inc., revised and enlarged edition, 1980.

Christian P. Robert. *The Bayesian Choice.* Springer Verlag, 2nd edition, 2001.

Ralph Tyrell Rockafellar. *Convex Analysis.* Princeton University Press, reprint edition, 1996.

Roman Rosipal and Leonard J. Trejo. Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of Machine Learning Research*, 2:97–123, 2001.

James Rovnyak. Methods of krein space operator theory. *Operator Theory Advances and Applications*, 134:31–66, 2002.

Yousef Saad. *Iterative Methods for Sparse Linear Systems.* Society for Industrial and Applied Mathematics, 2nd edition, 2000.

Saburou Saitoh. *Theory of Reproducing Kernels and its Applications.* Longman Scientific & Technical, 1988.

B. Schölkopf. The kernel trick for distances. In *Advances in Neural Information Processing Systems*, 2001.

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

Bernhard Schölkopf, Alex J. Smola, Robert C. Williamson, and Peter L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, 2000.

Bernhard Schölkopf, Alexander Smola, and Klaus Robert Müller. Kernel principal component analysis. In B. Schölkopf C. J. C. Burges and A. J. Smola, editors, *Advances in Kernel Methods–Support Vector Learning*, pages 327–352. The MIT Press, 1999.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels.* The MIT Press, 2002.

Laurent Schwartz. Sous espaces Hilbertiens d'espaces vectoriels topologiques et noyaux associés. *Journal d'Analyse Mathématique*, 13:115–256, 1964. in French.

M. Seeger. Relationships between gaussian processes, support vector machines and smoothing splines. Technical report, University of Edinburgh, 1999.

J. Shawe-Taylor, C. Williams, N. Cristianini, and J. Kandola. Eigenspectrum of the Gram matrix and its relationship to the operator eigenspectrum. In *ALT 02*, volume 2533 of *Lecture Notes in Computer Science*, pages 23–40, 2002.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

Jonathan Richard Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Carnegie Mellon University, Edition 1 1/4, August 1994.

Alex J. Smola, Zoltan L. Ovari, and Robert C. Williamson. Regularization with dot-product kernels. In *NIPS*, pages 308–314, 2000.

Alex J. Smola, Bernhard Schölkopf, and Klaus-Robert Müller. The connection between regularization operators and support vector kernels. *Neural Networks*, 11(4):637–649, 1998. URL `citeseer.nj.nec.com/smola98connection.html`.

Alexander J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. Technical Report NC2-TR-1998-030, ESPRIT Working Group in Neural and Computational Learning II, October 1998.

J.F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999. Special issue on Interior Point Methods (CD supplement with software).

Anrey N. Tikhonov and Vasiliy Y. Arsenin. *Solutions of ill-posed problems.* John Wiley & Sons, 1977.

Ivor W. Tsang and James T. Kwok. Efficient hyperkernel learning using second-order cone programming. In *European Conference on Machine Learning*, 2004.

Koji Tsuda, Shotaro Akaho, and Kiyoshi Asai. The EM algorithm for kernel matrix completion with auxiliary data. *Journal of Machine Learning Research*, 4:67–81, 2003.

Henk A. van der Vorst. Krylov subspace iteration. *Computing in Science and Engineering*, 2(1):32–37, 2000.

Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

Vladimir N. Vapnik. *The nature of statistical learning theory.* Springer Verlag, 1995.

Vladimir N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, 1998.

Vladmir N. Vapnik. *Estimation of Dependencies Based on Empirical Data.* Springer Verlag, 1982.

Grace Wahba. Ill posed problems: Numerical and statistical methods for mildly, moderately and severely ill posed problems with noisy data. Technical Report TR 595, University of Wisconsin-Madison Statistics Department, 1980.

Grace Wahba. Three topics in ill-posed inverse problems. In M. Engl and G. Groetsch, editors, *Inverse and Ill-Posed Problems*, pages 37–50. Academic Press, Inc., 1987.

Grace Wahba. *Spline Models for Observational Data.* Society for Industrial and Applied Mathematics, 1990.

Satosi Watanabe. *Pattern recognition: Human and Mechanical.* John Wiley & Sons, 1985.

Rüdiger Weiss. *Parameter-Free Iterative Linear Solvers*, volume 97 of *Mathematical Research.* Akademie Verlag, 1996.

C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo., editors, *Advances in Neural Information Processing Systems 8*, 1996.

Christopher K. I. Williams and David Barber. Bayesian classification with gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12): 1342–1351, 1998. URL `citeseer.nj.nec.com/williams98bayesian.html`.

David H. Wolpert. The supervised learning no-free-lunch theorems. In *Proceedings of the Sixth Online World Conference on Soft Computing in Industrial Applications*, 2001.

Laurent Zwald, Olivier Bousquet, and Gilles Blanchard. Statistical properties of kernel principal component analysis. In *Proceedings of the Conference on Learning Theory, 2004*, pages 594–608, 2004.