Spectral Properties of the Kernel Matrix and their Relation to Kernel Methods in Machine Learning

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Mikio Ludwig Braun

aus

Brühl, Rheinland

Bonn 2005

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn.
Diese Dissertation ist auf dem Hochschulschriftenserver der ULB Bonn `http://hss.ulb.uni-bonn.de/diss_online` elektronisch publiziert.

1. Referent: Prof. Dr. Joachim Buhmann
2. Referent: Prof. Dr. Michael Clausen
Tag der Promotionsprüfung: 27. Juli 2005
Erscheinungsjahr: 2005

Summary

Machine learning is an area of research concerned with the construction of algorithms which are able to learn from examples. Among such algorithms, so-called kernel methods form an important family of algorithms which have proven to be powerful and versatile for a large number of problem areas. Central to these approaches is the kernel matrix which summarizes the information contained in the training examples. The goal of this thesis is to analyze machine learning kernel methods based on properties of the kernel matrix. The algorithms considered are kernel principal component analysis and kernel ridge regression. This thesis is divided into two parts: a theoretical part devoted to studying the spectral properties of the kernel matrix, and an application part which analyzes the kernel principal component analysis method and kernel based regression based on these theoretical results.

In the theoretical part, convergence properties of the eigenvalues and eigenvectors of the kernel matrix are studied. We derive accurate bounds on the approximation error which have the important property that the error bounds scale with the magnitude of the eigenvalue, predicting correctly that the approximation error of small eigenvalues is much smaller than that of large eigenvalues. In this respect, the results improve significantly on existing results. A similar result is proven for scalar products with eigenvectors of the kernel matrix. It is shown that the scalar products with eigenvectors corresponding to small eigenvalues are small a priori independently of the degree of approximation.

In the application part, we discuss the following topics. For kernel principal component analysis, we show that the estimated eigenvalues approximate the true principal values with high precision. Next, we discuss the general setting of kernel based regression and show that the relevant information of the labels is contained in the first few coefficients of the label vector in the basis of eigenvectors of the kernel matrix, such that the information and the noise can be divided much more easily in this representation. Finally, we show that kernel ridge regression works by suppressing all but the leading coefficients, thereby extracting the relevant information of the label vectors. This interpretation suggests an estimate of the number of relevant coefficients in order to perform model selection. In an experimental evaluation, this approach proves to perform competitively to state-of-the-art methods.

Contents

Chapter 1

Introduction

Machine learning is an interdisciplinary area of research concerned with constructing machines which are able to learn from examples. One large class of tasks within machine learning is that of *supervised learning*. Here, a number of training examples is presented to the algorithm. These training examples consist of object features together with some label information which should be learned by the algorithm. The *classification* task consists in learning to correctly predict the membership of objects to one of a finite number of classes. If the label to be predicted is a real number, then this task is called *regression*.

So-called kernel methods are a class of algorithms which have proven to be very powerful and versatile for this type of learning problems. These methods construct the functional dependency to be learned by using kernel functions placed around each observation in the training set. There exist a large number of different variants of kernel methods, among them such prominent examples like the support vector machines and Gaussian processes regression.

Common to these methods is the use of a kernel function k, which assigns a real number to a given object pair. This number is typically interpreted as a measure of similarity between the objects. Central to kernel methods is the *kernel matrix*, which is built by evaluating k on all pairs of objects of the training set. Obviously, this matrix contains an exhaustive summary of the relationship between the objects as measured by the kernel function. In fact, for the training step of many algorithms, the object features are no longer necessary once the kernel matrix is computed.

For a certain class of kernel functions, so-called *Mercer kernels*, the kernel matrix is symmetric and positive definite. It is well known that such matrices have a particularly nice spectral decomposition, having a full set of eigenvectors which are orthogonal and only positive eigenvalues. This spectral decomposition characterizes the kernel matrix fully.

In this thesis, we will focus on two machine learning kernel algorithms, kernel principal component analysis and kernel ridge regression. Both are non-linear extension of classical methods from statistics. Principal component analysis is an unsupervised method which analyzes the structure of a finite data set in a vectorial setting. The result is a set of orthogonal directions along which the variance of the data is maximized. Kernel ridge regression is a non-linear extension of classical regularized least squares regression procedures. Kernel ridge regression has close connections to the Gaussian process method from the Bayesian inference framework. Kernel ridge regression and Gaussian processes have proven to perform competitively to support vector machines and can be considered state-of-the-art kernel methods for supervised learning. What distinguishes kernel ridge regression from support vector machines is that the learning step depends linearly on the labels and can be written in closed form using matrix algebra. For support vector machines, a quadratic optimization problem has to be solved, rendering the connection between the training examples and the computed solution less amenable to theoretical analysis. Moreover, the learning matrix is closely related to the kernel matrix, such that a detailed analysis of kernel ridge regression is closely related to the analysis of the spectral properties of the kernel matrix.

1.1 Goals of the Thesis

The main goal of this thesis is to perform an analysis of kernel principal component analysis and kernel ridge regression, which are both machine learning methods which can be described in terms of linear operators.

Generally speaking, current approaches to the analysis of machine learning algorithms tend to fall into one of the following two categories: Either the analysis is carried out in a fairly abstract setting, or the analysis appeals primarily to the intuition and to general principles considered to induce good learning behavior.

An example for the first case are consistency proofs of supervised learning algorithms based on capacity arguments, proving that the empirical risk converges to the true risk as the number of data points tends to infinity. These approaches often treat the algorithm as a black-box, reducing the algorithm to some opaque procedure which picks a solution from the so-called hypothesis space. While this technique has proven to be quite powerful and applicable to a large number of different settings, the resulting consistency proofs are sometimes lacking in some respect because they give no further insight into the exact mechanisms of the learning algorithm.

On the other hand, approaches of the second kind often lead to very nice interpretations, while these explanations often fail to translate into proofs of convergence. One could argue that the Bayesian framework sometimes falls into this category, since asymptotic considerations are often not included in the analysis of an algorithm. Instead, the usefulness of an algorithm is ensured by adhering to fundamental principles from the framework of Bayesian inference. Of course, full mathematical rigor is not a necessary requirement for machine learning algorithms. In fact, it is possible to write excellent introductions to the field without stating a single convergence proof (Hastie et al., 2001).

This thesis aims at bringing these two approaches closer together. An algorithm will not be treated as a black-box, but rather we will try to identify its essential components and then try to support experimental evidence with mathematical proofs. In the best case, the result will be explanations which have both, a certain intuitive appeal, and reliance on properties which are provably true. However, this does not mean that the mathematical tools which will be employed will be considerably less complex than in the black-box approach. But the results will make statements about components of the algorithms which will help to understand the algorithm.

The basic questions which will guide us are:

- *What is the structure of a finite data sample in relation to the kernel function employed?*

  A finite data set consists of a set of object samples and associated label informations (either categorical indices or real numbers). In a kernel method setting, this data set is implicitly mapped into a feature space in a non-linear fashion. We are interested in obtaining insight into the structure of the data set in feature space, for the object samples, and of the label information with respect to the object samples.

- *What kinds of performance guarantee can be given for kernel principal component analysis?*

  There has been an ongoing research effort to characterize the behavior of kernel principal component analysis for large sample sizes. We are interested in providing performance guarantees for the estimates of the principal values and the principal directions via the reconstruction error which are considerably more tight than existing results.

- *How does kernel ridge regression perform learning?*

  Consistency of kernel ridge regression has been proven via the theory of regularization networks (Evgeniou and Pontil, 1999), but we are interested in a more procedural explanation of how the computation of the fit function achieves learning.

- *How can the free parameters in kernel ridge regression be estimated effectively?*

  Kernel ridge regression requires the adjusting of the amount of regularization. We are interested if it is possible to estimate an effective choice for this regularization constant

based on insights into the structure of a data set. In particular, are sufficient structural insights into the label information available to not have to rely on hold-out testing?

1.2 Overview of the Thesis

The thesis is divided into two parts, a theoretic part discussing spectral properties of the kernel matrix, and an application part which discusses machine learning topics.

The first part of this thesis treats the spectral properties of the kernel matrix. As mentioned above, the kernel matrix is central to virtually any kernel algorithm and is therefore the first component we wish to study in detail. This area has been the focus of research for the last few years and we will improve upon existing results, providing bounds which correctly predict that the approximation error for small eigenvalues is much smaller than that for large eigenvalues, an effect which has so far not been modelled correctly.

The second part of this thesis is concerned with the analysis of machine learning algorithms, based on the results on the spectral properties of the kernel matrix. The first such application will be principal component analysis in both its traditional linear and in the kernelized version. We will be able to prove that the estimates of the principal decomposition converge with high accuracy.

In a supervised setting, we explore the relationship between the label information, which are the example values to be predicted, and the kernel matrix. The idea behind this approach is that independently of the learning algorithm used, the kernel function will be used to model the quantity which should be predicted. We will see that the vector of training labels has a specific structure with respect to the eigenvectors of the kernel matrix which allows us to easily isolate the information content in the labels.

Finally, we will turn to kernel ridge regression. This algorithm is an extension of the traditional linear ridge regression approach which computes a least squares fit while at the same time penalizing the length of the weight vector. Compared to other kernel algorithms, kernel ridge regression has the unique feature that the computation of the fit only involves the application of the inverse of a matrix. Moreover, the matrix which computes the fit is highly related to the kernel matrix since both have the same set of eigenvectors. As a result, the fit depends linearly on the training labels. This property should facilitate a theoretical analysis of the kernel ridge regression algorithm.

We give an overview of the original contributions developed in this thesis. The first part treats spectral properties of the kernel matrix.

Error Bounds for the Eigenvalues of the Kernel Matrix

Problem: We are interested in the exact structure of the eigenvalues of the kernel matrix. In particular, we want to explain the experimental findings that the eigenvalues decay as quickly as their asymptotic counterparts. This behavior is not implied by existing results, as these are either absolute error estimates or asymptotic central limit theorems.

Contribution of this work: A relative-absolute bound for the eigenvalues is derived which clearly shows that smaller eigenvalues have much smaller variance. This bound is significantly tighter than existing results. (Chapter 3)

Upper Bounds for Spectral Projections

Problem: We are interested in understanding how the scalar products between the eigenvectors of the kernel matrix and a subsampled smooth function behaves. Existing results are mainly asymptotic, showing that eventually, convergence takes place, but again experimental evidence suggests that the convergence is rather fast for certain eigenvectors, and happens on a scale relative to the magnitude of the eigenvalues.

Contribution of this work: An envelope on the scalar products with eigenvectors is derived. This is an upper bound which does not vanish asymptotically, but which is proportional to the

magnitude of the eigenvalue. This envelope shows that the scalar products between a function and eigenvectors are bounded by a constant times the associated eigenvalue plus a small error term.

This result displays a connection to the sampling theorem, stating that a smooth function has only limited complexity when subsampled at random points. (Chapter 4)

The second part of this thesis explores applications of these results to several kernel methods.

PCA, Kernel PCA and Finite-Sample Size Effective Dimension of Data in Feature Space

Problem: For principal component analysis (PCA), asymptotic results on the approximation error of the principal values have been known for some time (Anderson, 1963). We are interested in an estimate of the error for the non-asymptotic case. We are also interested in this question for kernel PCA, which is the kernelized version of PCA. This algorithm has been the focus of recent research which aims at specifying how convergence of kernel PCA should be formalized and how it can be proven. Since PCA is an unsupervised method, it is not directly evident how convergence should be formalized.

Since kernel PCA effectively analyzes the structure of data in feature space, results on kernel PCA give insights into the finite-sample structure of data in feature space. The structure of the feature space is usually not made explicitly, because the feature map is only given implicitly via the kernel function. For a finite sample, the question is if it is possible to make a statement about its structure. Trivially, a sample of size n is contained in an n dimensional subspace spanned by the sample points, but does the sample occupy this space evenly, or is it contained in some subspace of fixed dimension? An answer to this question can give some insight into the hardness of learning in an infinite-dimensional feature space.

Contribution of this work: The eigenvalue and eigenvector results directly translate to convergence results for PCA and kernel PCA. For PCA, we obtain a purely relative bound which shows that the approximation error scales with the eigenvalues. For kernel PCA, we obtain a relative-absolute bound, which consists of a relative term and a typically small absolute error term.

The consequences of these results for the data in feature space are interesting: It turns out that similar to the asymptotic case, the data is contained in an effectively finite-dimensional subspace of the (often infinite-dimensional) subspace. This can be thought of as a more direct statement of the well-known facts that large margin classifiers give rise to a class with finite VC-dimension, and that the fat-shattering dimension is finite. For the practitioner, these results mean that a finite data sample is in fact contained in a subspace with a fixed small dimension which does not scale with sample size. Therefore, even if the feature space is in principle infinite-dimensional, learning has to consider only a low-dimensional subspace of the feature space. (Chapter 5)

Significant Dimension of Data Given Label Information

Problem: Consider the following two problems:

(1) Irrespective of the training method employed, a kernel method for regression constructs the resulting fit function from kernels placed at the individual observations. Thus the relation between the label information and the kernel matrix forms sort of an *a priori* condition of the learning problem. We are interested in characterizing this relation.

(2) PCA is often used as de-noising step before performing the actual classification step. How many dimensions should be retained given that one wants to reconstruct a certain function encoded in a noise label vector Y? In the standard setting of vectorial data in a finite dimensional space, based on certain modelling assumptions, one can show that the principal values of the data exhibit some data-dependent decay of the principal values which then eventually makes a transition into a ramp with small slope which can be attributed to measurement noise. The standard approach to estimate the number of relevant directions analyzes the sequence of principal values to identify these noise directions. The simplest such method looks for a "knee" in the data. For kernel PCA, these modelling assumptions do not hold, such that the standard approach cannot be applied.

The question is if one can nevertheless estimate the number of relevant dimensions based on the additional information contained in the target labels Y.

Contribution of this work: The results on spectral projections give a very interesting answer to both these questions. We have seen that a smooth function will have rapidly decaying coefficients. On the other hand, we see that noise has evenly distributed coefficients. Therefore, we can estimate the number of dimensions which (1) should be reconstructed in learning, or (2) retained for de-noising effectively. We propose two such methods.

This result states that the interesting part of the label information is also contained in a finite-dimensional set, such that we finally see that the whole learning problem in feature space is inherently finite-dimensional, which also explains its success in practical applications. Put more directly, when using kernel methods, there is no curse of dimensionality as often stated, because the relevant part of the data is contained in an essentially finite dimensional subspace of the feature space. One only has to ensure that the algorithm is guarded against overfitting to the noise which is contained in the infinite dimensions. Protection against overfitting is achieved by regularization. (Chapter 6)

Analysis of the Fit Matrix in Kernel Ridge Regression

Problem: Kernel ridge regression (KRR) is a standard kernel method for regression and classification which has proven to work well. KRR is special in the sense that the in-sample fit is computed by applying a matrix to the label vector. Analyzing this matrix, it should be possible to understand KRR on a fairly detailed level.

Contribution of this work: The matrix is closely related to the kernel matrix. In fact, it is diagonal with respect to the basis of eigenvectors of the kernel matrix. KRR consists of three steps which can be readily understood using the other results of this thesis. We see that KRR effectively reconstructs the information contained in the first few coefficients of the spectrum of the label vector while noise is suppressed. This gives an alternative analysis of kernel ridge regression which is formally well-founded and also coincides with the intuitive ideas practitioners have in conjunction with KRR. (Chapter 7)

Estimating The Regularization Constant

Problem: Kernel ridge regression has two free parameters: The choice of the kernel and the choice of the regularization parameter. These are estimated either by estimating the generalization error by penalty terms or hold-out testing, or by performing maximum likelihood estimates in the context of Gaussian processes. Can we use the insights we have gained so far do part of the model choice without doing neither, hold-out testing nor maximum likelihood estimates?

Contribution of this work: Based on the procedures to estimate the dimensionality of the data, a heuristic to set the regularization constant is proposed. It is shown that this procedure leads to a rather effective procedure which displays that the insights obtained so far can actually be put to good use. (Chapter 7)

With respect to the layout, note that sections which discuss related work or existing results are set in a different font to set these sections off from the original content of this thesis.

1.3 Final Remarks

One of the main challenges of this work has been the conflict between the applied nature of the field of research of machine learning and the goal of this thesis to provide rigorous theoretical insights. This conflict manifests itself in several ways.

First of all, the more technical chapters might not be very accessible to non-technically inclined readers. I have tried to increase the accessibility by framing each chapter in an abstract, an introduction and a less technical statement of the main results. Each chapter is moreover ended

with a conclusion. These sections alone should give sufficient information on the content of the chapter and its relation to other parts of the thesis and machine learning in general.

Another problem is that it is impossible to maintain a pure level of mathematical rigor throughout the thesis. Many approaches and insights are of an intuitive nature which has not yet found its way into a mathematical formulation. One cannot help to notice that the theoretical underpinnings of many practices in machine learning are insufficient and that one has to step beyond what is known and proven and let oneself be guided by one's own intuition to understand problems. The consequences for this thesis is that especially in the later chapters arguments become less rigorous, and results are no longer presented as theorems but are developed in the main text. Here, a problem occurs if one uses one of the theorems from the technical chapters (notably Chapter 3 and 4) in an argument, because there is really no point in citing the theorem with the full level of mathematical detail if the argument itself is not on the same level. Therefore, theorems are often only cited with respect to their intuitive content. One should nevertheless keep in mind that these theorems are actually rigorous.

Finally, it seems that the standards for theoretical results are higher in applied fields. This is because researchers in this field often have a quite good grasp of the properties of the algorithms with which they work every day. Therefore, a theoretical result which does not manage to describe the well-known properties of these algorithms at least to some extent is usually put down as being just that, a theoretical result with lack of relevance. Therefore, in this thesis, a lot of effort has been put into deriving, for example, error bounds which show the same behavior as observable in numerical simulations. In fact, convergence of the eigenvalues has also already been known, but existing bounds failed to describe the behavior of the approximation errors accurately. Furthermore, in this thesis, the theoretical results have often be accompanied with plots of numerical simulations in order to show that the theoretically predicted behavior matches the actual behavior.

Chapter 2

Preliminaries

Abstract

This chapter serves as a brief introduction to the supervised learning setting and kernel methods. Moreover, several results from linear algebra, probability theory, and functional analysis are reviewed which will be used throughout the thesis.

2.1 Some notational conventions

We begin by introducing some basic notational conventions. The sets $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{C}$ denote the natural, integer, real, and complex numbers.

Vectors will be denoted by lowercase letters, whereas matrices will be denoted by bold uppercase letters. Random variables will be denoted by uppercase letters. The individual entries of vectors and matrices are denoted by square brackets. For example, $x \in \mathbb{R}^n$ is a vector with coefficients $[x]_i$. The matrix $\mathbf{A}$ has entries $[\mathbf{A}]_{ij}$. Vector and matrix transpose is denoted by $x^\top$. Sometimes, the set of square $n \times n$ matrices are denoted by $\mathbb{M}_n$, and the set of general $n \times m$ matrices by $\mathbb{M}_{n,m}$.

The set of eigenvalues of a square matrix $\mathbf{A}$ is denoted by $\lambda(\mathbf{A})$. For a symmetric $n \times n$ matrix $\mathbf{A}$, we will always assume that the eigenvalues and eigenvectors are sorted in non-increasing order with eigenvalues repeated according to their multiplicity. The eigenvalues of $\mathbf{A}$ are thus $\lambda_1(\mathbf{A}) \geq \ldots \geq \lambda_n(\mathbf{A})$.

We use the following standard norms on finite-dimensional vector spaces. Let $x \in \mathbb{R}^n$ and $\mathbf{A} \in \mathbb{M}_n$. Then,

$$\|x\| = \sqrt{\sum_{i=1}^{n} [x]_i^2}, \qquad \|\mathbf{A}\| = \max_{x\colon \|x\| \neq 0} \frac{\|\mathbf{A}x\|}{\|x\|}. \tag{2.1}$$

A useful upper bound on $\|\mathbf{A}\|$ is given by

$$\|\mathbf{A}\| \leq n \max_{1 \leq i,j \leq n} |[\mathbf{A}]_{ij}|. \tag{2.2}$$

Another matrix norm we will encounter is the Frobenius norm

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i,j=1}^{n} [\mathbf{A}]_{ij}^2}. \tag{2.3}$$

As usual, δ_{ij} denotes the Kronecker delta which is equal to 1 if $i = j$ and 0 else.

Frequently used symbols are summarized in the symbol table on page 14.

$\mathcal{X}$	space of object samples
$\mathcal{Y}$	space of label samples
μ	probability measure on $\mathcal{X}$
$\mathcal{H}_\mu(\mathcal{X})$	Hilbert space of functions on $\mathcal{X}$ (p. 16)
P	probability
E, E_μ	expectation (with respect to measure μ)
Var_μ	variance (with respect to measure μ)
n	sample size
$X_1, \ldots, X_n$	object samples
$Y_1, \ldots, Y_n$	label samples
$\mathbf{X}$	matrix whose columns are the object samples
$f(\mathbf{X})$	sample vector $f(\mathbf{X}) = (f(X_1), \ldots, f(X_n))^\top$.
k	Mercer kernel function (p. 17)
T_k	integral operator associated with k (p. 17)
λ_i	eigenvalues of k
ψ_i	eigenfunctions of k
r	truncation points
$k^{[r]}$	truncated kernel function (p. 18)
e^r	truncation error function $k - k^{[r]}$
$f^{[r]}$	truncated function
$\mathbf{K}_n$	(normalized) kernel matrix (p. 18)
l_i, u_i	eigenvalues, eigenvectors of $\mathbf{K}_n$
$\mathbf{K}_n^{[r]}$	truncated kernel matrix
m_i, v_i	eigenvalues, eigenvectors of $\mathbf{K}_n^{[r]}$
$\mathbf{E}_n^r$	truncation error matrix $\mathbf{K} - \mathbf{K}^{[r]}$
$\mathbf{\Psi}_n^r$	relative error matrix (p. 37)
$C(r, n)$	relative error term (p. 34)
$E(r, n)$	absolute error term (p. 34)
$T(r, n)$	function truncation error term (p. 75)
$\Lambda_{>r}$	sum of eigenvalues smaller than λ_r

Figure 2.1: Symbol table

2.2 Probability Theory

Since we will usually consider subsets of $\mathbb{R}^d$ as probability spaces, we will quietly assume the associated Borel σ-algebra, meaning that topological sets are measurable. Therefore, all closed sets, open sets, point sets, and countable combinations of those will be measurable, which is enough for our purposes. Let $\mathcal{X} \subseteq \mathbb{R}^d$, $\mathcal{X}$ measurable. For the following, let μ be a probability measure on $\mathcal{X}$. A special probability measure is the Dirac measure δ_x, for some $x \in \mathcal{X}$. It represents a point-mass at x, which means that $\delta_x(A) = 1$ if and only if $x \in A$.

The expectation of a random variable $X\colon \mathcal{X} \to \mathbb{R}$ will be denoted by $\mathbf{E}_\mu(X)$, its variance by $\mathbf{Var}_\mu(X)$. If the probability measure μ is not specified, a generic measure $\mathbf{P}$ will be assumed which is defined on some probability space $(\Omega, \mathcal{A}, \mathbf{P})$ which is sufficiently rich to allow us to construct all random variables which will be considered. The notation $X \sim \mu$ means that X is distributed as μ, and $\mathbf{P}_X$ is the distribution[1] of X, such that trivially, $X \sim \mathbf{P}_X$. Using the Dirac measure, one can write the empirical distribution μ_n of an i.i.d. (independent and identically distributed) sample $X_1, \ldots, X_n$ with common distribution μ as

$$\mu_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{X_i}. \tag{2.4}$$

The (multi-variate) Gaussian distribution (or Normal distribution) is the probability measure in $\mathbb{R}^d$ with probability density

$$p(x) = (2\pi)^{-\frac{1}{2}d} (\det \mathbf{\Sigma})^{-\frac{1}{2}} \exp\left(-\frac{1}{2} (x-\mu)^\top \mathbf{\Sigma}^{-1} (x-\mu) \right), \tag{2.5}$$

where $\mathbf{\Sigma}$ is the covariance matrix, and μ is the mean vector. For $d = 1$, the formula becomes

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{|x-\mu|^2}{2\sigma^2} \right). \tag{2.6}$$

2.3 Learning Settings

The supervised learning setting is usually formalized as follows (compare (Devroye et al., 1996)): The object features are assumed to lie in some space $\mathcal{X}$ while the labels lie in some space $\mathcal{Y}$. The training examples are generated as i.i.d. samples from a probability distribution $\mathbf{P}_{\mathcal{X}\times\mathcal{Y}}$. A training set of size n is then given as $(X_1, Y_1), \ldots, (X_n, Y_n)$, where $(X_i, Y_i) \sim \mathbf{P}_{\mathcal{X}\times\mathcal{Y}}$.

One distinguishes two types of supervised learning problems, depending on the structure of $\mathcal{Y}$. For *classification*, $\mathcal{Y}$ consists of a finite number of class labels, and the task is to correctly predict the class membership of objects. For *regression*, $\mathcal{Y} = \mathbb{R}$, and the task is to predict some real quantity based on the object features.

A learning algorithm has the task to take such a training set of size n and to output a result which allows to make predictions for new objects. This output of a learning algorithm is a mapping $g\colon \mathcal{X} \to \mathcal{Y}$ which is called *fit function* (for regression), or *classifier* (for classification). Let us call the output *predictor* when we do not want to specify the type of supervised problem we are addressing.

The quality of a predictor g is measured as the *expected error* of the predicted labels, sometimes also called the *generalization error*. For that, we need a *loss function* $\mathfrak{L}$ on $\mathcal{Y}$. This is a function $\mathfrak{L}\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. If $(X, Y) \sim \mathbf{P}_{\mathcal{X}\times\mathcal{Y}}$, the expected error is given as

$$\mathbf{E}(\mathfrak{L}(g(X), Y)). \tag{2.7}$$

The standard choice for classification is the 0-1–loss error

$$\mathfrak{L}_{0\text{-}1}(y, y') = \begin{cases} 1 & y \neq y' \\ 0 & \text{else.} \end{cases} \tag{2.8}$$

[1] In this thesis, "distribution" will be used synonymously for "measure".

One can easily compute that in this case

$$\mathbf{E}(\mathfrak{L}_{0\text{-}1}(g(X), Y)) = \mathbf{P}\{g(X) \neq Y\}, \tag{2.9}$$

the probability to make an incorrect prediction. The optimal prediction is given by assigning X to the label which is most probable. The associated minimal expected error is called the *Bayes risk*.

For regression, the standard choice is the squared error $\mathfrak{L}_2$,

$$\mathfrak{L}_2(y, y') = (y - y')^2. \tag{2.10}$$

For regression, one frequently uses the following modelling assumption: One assumes that there exists a *target function* $f\colon \mathcal{X} \to \mathbb{R}$, whose measurements are contaminated by additive zero-mean noise:

$$Y = f(X) + \varepsilon_X, \tag{2.11}$$

where f is the *target function*, and $(\varepsilon_x)_{x\in\mathcal{X}}$ is a family of independent zero mean random variables. One can show that the optimal solution is given as

$$g(x) = \mathbf{E}(Y|X = x), \tag{2.12}$$

and it holds that $\mathbf{E}(Y|X = x) = f(x)$. In this case, the Bayes risk is given by $\mathbf{Var}(\varepsilon_X)$, the variance of the noise randomly selected according to X.

2.4 Kernel Functions

This section serves mainly to introduce Mercer kernel functions. Usually, Mercer kernels are introduced as symmetric functions on $\mathcal{X}$ which obey some form of positivity condition, which is difficult to prove in general. Then, it is proven that these functions have a certain type of infinite expansion known as Mercer's formula (see below).

In this thesis, we will take the opposite approach and define Mercer kernels starting with a ℓ^1 sequence of real numbers and a set of orthogonal functions using Mercer's formula. The advantage of this approach is that the relation between the kernel function and a specific choice of eigenvalues and eigenfunctions is made explicit. In the usual setting, Mercer's theorem (see below) ensures the existence of such eigenvalues and eigenfunctions allowing an expansion as will be introduced below, but there can exist more than one such choice.

To begin with, we need a Hilbert space to define the notion of orthogonality. First of all, define the scalar product between two measurable functions $f, g\colon \mathcal{X} \to \mathbb{R}$ as

$$\langle f, g\rangle_\mu = \int_{\mathcal{X}} f(x)g(x)\mu(dx), \tag{2.13}$$

where μ is the marginal distribution of $\mathbf{P}_{\mathcal{X}\times\mathcal{Y}}$ on $\mathcal{X}$. The norm of f is defined as $\|f\| = \sqrt{\langle f, f\rangle}$. In principle, we want to consider the space of functions with finite norm. However, since one can modify f on a set of measure zero without changing its norm, we have to identify functions f, g with $\|f - g\| = 0$. Let $\sim$ be the equivalence relation such that $f \sim g$ if and only if $\|f - g\| = 0$. The Hilbert space we will use throughout this thesis is then given as the set

$$\mathcal{H}_\mu(\mathcal{X}) = \{f\colon \mathcal{X} \to \mathbb{R} \mid \langle f, f\rangle_\mu < \infty\}/\sim \tag{2.14}$$

equipped with scalar product $\langle \cdot, \cdot \rangle_\mu$. As usual, two functions $f, g \in \mathcal{H}_\mu(\mathcal{X})$ are orthogonal if $\langle f, g\rangle_\mu = 0$.

Strictly speaking, the elements of $\mathcal{H}_\mu(\mathcal{X})$ are equivalence classes of functions. However, we will usually speak of these equivalence classes as functions in order to reduce the notational overhead. A crucial difference must not be forgotten, though: since the elements of $\mathcal{H}_\mu(\mathcal{X})$ are equivalence classes of functions, point evaluations are not well-defined if point sets have measure zero because

the different representatives of an equivalence class $f \in \mathcal{H}_\mu(\mathcal{X})$ may differ on sets of measure zero. However, a different situation is given by random point evaluations $f(X)$ with X being distributed as μ. In this case, only the distribution of $f(X)$ is relevant, and this distribution is the same for different representatives precisely because the sets on which two representatives differ have measure zero at most.

Note that the scalar product is defined with respect to the same probability measure which generates the object samples. It turns out that this approach is necessary to obtain the correct convergence relationships for the eigenvalues and eigenvectors of the kernel matrix (see Chapters 3 and 4). This choice for μ is also more compatible with non-compact domains $\mathcal{X}$. Even if the object space $\mathcal{X}$ is unbounded, using a finite measure ensures that integral operators (see below) have always a discrete spectrum. This is not the case if one uses for example the ordinary Lebesgue measure which can result in Mercer kernels with continuous parts of the spectrum whose treatment is considerably more involved (see for example Williamson et al. (2001)).

Mercer kernels are then defined as follows:

Definition 2.15 (Mercer kernel) Let μ be a probability measure on $\mathcal{X}$, and $\mathcal{H}_\mu(\mathcal{X})$ the associated Hilbert space. Given a sequence $(\lambda_i)_{i\in\mathbb{N}} \in \ell^1$ with $\lambda_i \geq 0$, and an orthogonal family of unit norm functions $(\psi_i)_{i\in\mathbb{N}}$ with $\psi_i \in \mathcal{H}_\mu(\mathcal{X})$, the associated Mercer kernel is

$$k(x,y) = \sum_{i=1}^{\infty} \lambda_i \psi_i(x)\psi_i(y). \tag{2.16}$$

The numbers λ_i will be called the eigenvalues of the kernel and ψ_i its eigenfunctions.

Contrary to what is often stated in conjunction with the Mercer formula, the series need not converge uniformly over $\mathcal{X} \times \mathcal{X}$, for example for non-continuous functions ψ_i.

In practical situations, one will often use a Mercer kernel function where the above sum can be computed in closed form, and the expansion itself is in fact unknown. An example for such a kernel function is the radial basis kernel function (rbf-kernel):

$$k_{\mathrm{rbf}}(x,y) = \exp\left(-\frac{\|x-y\|^2}{2w}\right) \tag{2.17}$$

which is parameterized by the kernel width $w > 0$.

In this thesis, we will mostly focus on kernels which have an infinite expansion and are moreover uniformly bounded. This ensures a certain degree of regularity of the eigenfunctions. Kernels which have a known finite expansion like for example polynomial kernels are already well understood by explicitly writing down the expansion. For kernels with infinite dimensional expansions, this is not possible for obvious reason. However, one can show that the eigenvalues and eigenvectors of the associated kernel matrix to be introduced below approximate the true eigenvalues and eigenfunctions. This relationship allows to study the properties of kernels with infinite expansions.

We have called the numbers λ_i eigenvalues and the functions eigenfunctions ψ_i. Actually, these are the eigenvalues and eigenfunctions of the *integral operator associated with* k defined by

$$T_k f(x) = \int_{\mathcal{X}} k(x,y) f(y) \mu(dy). \tag{2.18}$$

Lemma 2.19 *The λ_i and ψ_i occurring in the definition of a uniformly bounded Mercer kernel function k are the eigenvalues and eigenfunctions of T_k.*

Proof We compute $T_k\psi_i$(x):

$$\begin{aligned} T_k\psi_i(x) = \int_{\mathcal{X}} k(x,y)\psi_i(y)\mu(dy) &= \int_{\mathcal{X}} \sum_{j=1}^{\infty} \lambda_j \psi_j(x)\psi_j(y)\psi_i(y)\mu(dy) \\ &= \sum_{j=1}^{\infty} \lambda_j \psi_j(x)\langle\psi_j,\psi_i\rangle = \lambda_i\psi_i(x), \end{aligned} \tag{2.20}$$

where the integral and the summation exchange due to the boundedness of k. ■

We will often approximate a Mercer kernel function using only a finite number of terms in Mercer's formula (2.16):

Definition 2.21 (Degenerate Mercer kernel function and r-degenerate approximation)
A Mercer kernel function k on $\mathcal{H}_\mu(\mathcal{X})$ is called *degenerate*, if it has only a finite number of non-zero eigenvalues. The *r-degenerate approximation* $k^{[r]}$ of a kernel function k with eigenvalues (λ_i) and eigenfunctions (ψ_i) is defined as

$$k^{[r]}(x,y) = \sum_{i=1}^{r} \lambda_i \psi_i(x)\psi_i(y). \tag{2.22}$$

Note that an r-degenerate approximation does not necessarily have r non-zero eigenvalues, because some of the initial r eigenvalues can be zero as well. But for our purposes, this definition is sufficient and it is not necessary to devise a more elaborate definition.

The *kernel matrix* is the square matrix obtained by evaluating the kernel function on all pairs of object samples X_i, X_j. In other words, the normalized kernel matrix $\mathbf{K}_n$ is the $n \times n$ square matrix with entries

$$[\mathbf{K}_n]_{ij} = \frac{1}{n} k(X_i, X_j). \tag{2.23}$$

Accordingly, we will consider approximations based on the r-degenerate approximation of the kernel function,

$$[\mathbf{K}_n^{[r]}]_{ij} = \frac{1}{n} k^{[r]}(X_i, X_j). \tag{2.24}$$

In Chapter 3 and 4, we will study the eigenvalues and eigenvectors of the kernel matrix in depth.

We close this section with some remarks concerning integral operators. These are closely related to kernel functions as we have already seen in Lemma 2.19.

We begin by reviewing some fundamental results on self-adjoint compact operators. First of all, since integral operators act on a function space, we formally introduce such a space as a Banach space $(\mathcal{B}, \|\cdot\|)$, which is a (possible infinite-dimensional) complete vector space with a norm. A *compact* operator is an operator which maps bounded sets to compact sets. Recall that compact sets are sets such that any open covering has a finite sub-covering. This fact can be interpreted as compact sets having finite complexity at any given finite scale. Typical examples for compact operators are integral operators which we have already introduced in (2.18).

An important property of compact operators is that they have at most countably infinitely many non-zero eigenvalues. Moreover, the only near point of the set of eigenvalues is 0. If, in addition, we are able to define a scalar product on $\mathcal{B}$ which induces the norm via $\|\cdot\| = \langle\cdot\,,\cdot\rangle^{1/2}$, the Banach space becomes a Hilbert space $\mathcal{H}$. An operator T is called *self-adjoint* if for all $x, y \in \mathcal{H}$,

$$\langle x, Ty\rangle = \langle Tx, y\rangle. \tag{2.25}$$

Self-adjoint compact operators are of special interest because they only have real eigenvalues, and eigenfunctions to different eigenvalues are orthogonal. For such operators we obtain the following useful result:

Theorem 2.26 (Spectral decomposition theorem for self-adjoint operators) *Let T be a self-adjoint compact operator on a Hilbert space $\mathcal{H}$. Then there exists an orthonormal family of functions $(\psi_i)_{i\in\mathbb{N}}$ and a null sequence $(\lambda_i)_{i\in\mathbb{N}}$, such that for all $f \in \mathcal{H}$,*

$$Tf = \sum_{i=1}^{\infty} \lambda_i \psi_i \langle \psi_i, f\rangle. \tag{2.27}$$

In addition, for Mercer kernels, we also know that the eigenvalues are positive and summable, which proves to be very useful.

Certain properties of the kernel function are inherited by the image under an integral operator. We collect some of these in the following lemmas. We will assume throughout that μ is a probability measure and the Hilbert space is $\mathcal{H}_\mu(\mathcal{X})$ as introduced in (2.14). The $\|\cdot\|_p$ norms are defined analogously as

$$\|f\|_p = \left(\int_{\mathcal{X}} |f|^p \, d\mu\right)^{1/p}. \tag{2.28}$$

If p is not explicitly specified, the default $p = 2$ is assumed. An important inequality is the Hölder inequality

$$|\langle f, g\rangle| \leq \|f\|_p \|g\|_q, \qquad \frac{1}{p} + \frac{1}{q} = 1. \tag{2.29}$$

Lemma 2.30 (Boundedness of images under T_k)
Let $K = \sup_{x,y\in\mathcal{X}} |k(x,y)|$, then $|T_k f(x)| \leq K\|f\|$ for all $x \in \mathcal{X}$.

Proof Let $k_x(y) = k(x,y)$. By the definition of k, we readily see that k_x is measurable. Moreover, since μ is a probability measure, $\|k_x\| \leq K$, and consequently, $k_x \in \mathcal{H}_\mu(\mathcal{X})$. Now, applying the Hölder inequality with $p = \frac{1}{2}$:

$$|T_k f(x)| = \left|\int k(x,y) f(y) \mu(dy)\right| \leq \|k_x\| \|f\| \tag{2.31}$$

proves the lemma. ■

Note that since, strictly speaking, k is only known up to modifications on sets of measure zero, the supremum in the last lemma has to be interpreted as an essential supremum.

The next lemma treats Lipschitz continuity. Let $\mathcal{X}$ be equipped with a norm. We say that a function $f\colon \mathcal{X} \to \mathbb{R}$ is Lipschitz continuous with constant L, if L is the smallest number such that for all $x, y \in \mathcal{X}$,

$$|f(x) - f(y)| \leq L\|x - y\|. \tag{2.32}$$

We consider a kernel function which is uniformly Lipschitz continuous in the first argument. Since, again strictly speaking, k is known only up to modifications on sets of measure zero, we will assume that there exists a Lipschitz-continuous representative of k, which is considered in the following.

Lemma 2.33 (Lipschitz continuity of images under T_k)
Let k be a kernel function such that there exists an L such that for all $x, x' \in \mathcal{X}$,

$$\sup_{y\in\mathcal{X}} |k(x,y) - k(x',y)| \leq L\|x - x'\|. \tag{2.34}$$

Then, $T_k f$ is Lipschitz continuous with a constant $\leq L\|f\|$.

Proof Using the Hölder inequality with $p = \frac{1}{2}$, it follows that

$$|T_k f(x) - T_k f(x')| = \left|\int \big(k(x,y) - k(x',y)\big) f(y) \mu(dy)\right| \leq \|k_x - k_{x'}\| \|f\|. \tag{2.35}$$

Now, since μ is a probability measure, $\mu(\mathcal{X}) = 1$, and

$$\|k_x - k_{x'}\| \leq \sup_{y\in\mathcal{X}} |k(x,y) - k(x',y)| \mu(\mathcal{X}) = \sup_{y\in\mathcal{X}} |k(x,y) - k(x',y)| \leq L\|x - x'\|. \tag{2.36}$$

Thus,

$$|T_k f(x) - T_k f(x')| \leq L\|x - x'\| \|f\|, \tag{2.37}$$

and $T_k f$ is Lipschitz continuous with a constant which is at most as large as $L\|f\|$. ■

Therefore, the Lipschitz constant can be considered as a measure of regularity of the kernel function, and $T_k f$ is as smooth as L times the norm of f.

2.5 Large Deviation Bounds

We collect some large deviation bounds for sums of independent random variables. The most basic one is the Chebychev-inequality which is known to provide rather loose bounds but which will nevertheless be of good use for us.

Theorem 2.38 (The Chebychev inequality) *(Bauer, 1990) Let $X_1, \ldots, X_n$ be i.i.d. random variables with $\mathbf{E}(X_1) = 0$ and $\mathbf{Var}(X_1) = \sigma^2 < \infty$. Then,*

$$\mathbf{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n} X_i\right| \geq \varepsilon\right\} \leq \frac{\sigma^2}{n\varepsilon^2}. \tag{2.39}$$

The following result bounds the large deviation probability based on the size of the range of the random variables.

Theorem 2.40 (The Hoeffding inequality) *(Hoeffding (1963), also Steele (1997))*
Let $X_1, \ldots, X_n$ be i.i.d. random variables with zero mean and $|X_i| \leq M < \infty$. Then,

$$\mathbf{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n} X_i\right| \geq \varepsilon\right\} \leq 2\exp\left(-\frac{2n\varepsilon^2}{M^2}\right). \tag{2.41}$$

The next result also considers the variance of the random variable leading to better results under certain conditions.

Theorem 2.42 (The Bernstein inequality) *(van der Vaart and Wellner, 1998)*
Let $X_1, \ldots, X_n$ be i.i.d. random variables with zero mean and $|X_i| \leq M < \infty$ and $\mathbf{Var}(X_i) = \sigma^2 < \infty$. Then,

$$\mathbf{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n} X_i\right| \geq \varepsilon\right\} \leq 2\exp\left(-\frac{1}{2}\frac{n\varepsilon^2}{\sigma^2 + \frac{M\varepsilon}{3}}\right) \tag{2.43}$$

From each of these large deviation bounds, one can derive a bound on the deviation given a certain confidence $0 < \delta < 1$.

Theorem 2.44 *Let X_i be i.i.d. samples with $\mathbf{E}(X_i) = 0$, and $\mathbf{Var}(X_i) = \sigma^2$, $|X_i| \leq M < \infty$. Then, for $S_n = \frac{1}{n}\sum_{i=1}^{n} X_i$, it holds that with probability larger than $1 - \delta$,*

$$|S_n| < \sqrt{\frac{\sigma^2}{n\delta}}, \tag{2.45}$$

$$|S_n| < M\sqrt{\frac{1}{2n}\log\frac{2}{\delta}}, \tag{2.46}$$

$$|S_n| < \frac{2M}{3n}\log\frac{2}{\delta} + \sqrt{\frac{2\sigma^2}{n}\log\frac{2}{\delta}}. \tag{2.47}$$

Proof Inequality (2.45) follows from the Chebychev inequality:

$$\mathbf{P}\{|S_n| \geq \varepsilon\} \leq \frac{\sigma^2}{n\varepsilon^2} = \delta, \qquad \Rightarrow \qquad \varepsilon = \sqrt{\frac{\sigma^2}{n\delta}}. \tag{2.48}$$

Therefore, for $\varepsilon = \sqrt{\sigma^2/n\delta}$,

$$\mathbf{P}\left\{|S_n| < \sqrt{\frac{\sigma^2}{n\delta}}\right\} \geq 1 - \delta. \tag{2.49}$$

Inequality (2.46) follows from the Hoeffding inequality (2.41):

$$\mathbf{P}\{|S_n| \geq \varepsilon\} \leq 2\exp\left(-\frac{n\varepsilon^2}{2M}\right) = \delta \tag{2.50}$$

and solving for ε.

Finally, (2.47) follows from the Bernstein inequality:

$$\mathbf{P}\{|S_n| \geq \varepsilon\} \leq 2\exp\left(-\frac{1}{2}\frac{n\varepsilon^2}{\sigma^2 + \frac{M\varepsilon}{3}}\right). \tag{2.51}$$

Setting the right hand side to δ and partially solving for ε results in

$$2\log\frac{2}{\delta} = \frac{n\varepsilon^2}{\sigma^2 + \frac{M}{3}\varepsilon}. \tag{2.52}$$

We use the abbreviation $2\log\frac{2}{\delta} = d$. With that, the last display is equivalent to

$$\sigma^2 d + \frac{M}{3}\varepsilon d = n\varepsilon^2, \tag{2.53}$$

which finally results in the quadratic equation

$$0 = \varepsilon^2 - \frac{Md}{3n}\varepsilon - \frac{\sigma^2 d}{n}. \tag{2.54}$$

This equation has the two solutions

$$\varepsilon_\pm = \frac{Md}{6n} \pm \sqrt{\frac{M^2d^2}{36n^2} + \frac{\sigma^2 d}{n}}. \tag{2.55}$$

The solution with the minus in front of the square root is negative, therefore, the solution is ε_+. Since for $a, b \geq 0$, $\sqrt{a^2+b^2} \leq a + b$, we get a more convenient upper bound on ε_+:

$$\varepsilon_+ \leq \frac{Md}{3n} + \sqrt{\frac{\sigma^2 d}{n}}. \tag{2.56}$$

Substituting the definition of d gives (2.47). ■

Finally, a confidence bound for the sum of two random variables can be easily constructed from individual confidence bounds.

Lemma 2.57 (Combining Large Deviation Bounds) *Let X, X' be positive random variables such that*

$$\mathbf{P}\{X > \varepsilon\} \leq \delta, \qquad \mathbf{P}\{X' > \varepsilon'\} \leq \delta. \tag{2.58}$$

Then,

$$\mathbf{P}\{X + X' > \varepsilon + \varepsilon'\} \leq 2\delta. \tag{2.59}$$

Proof Note that

$$\begin{aligned}\mathbf{P}\{X + X' > \varepsilon + \varepsilon'\} &\leq \mathbf{P}\{X > \varepsilon \text{ or } X' > \varepsilon'\} \\ &\leq \mathbf{P}\{X > \varepsilon\} + \mathbf{P}\{X' > \varepsilon'\} \leq 2\delta.\end{aligned} \tag{2.60}$$

■

Part I

Spectral Properties of the Kernel Matrix

Chapter 3

Eigenvalues

Abstract

The subject of this chapter are the eigenvalues of the kernel matrix. We derive bounds on the approximation error for the non-asymptotic case. The resulting error bounds are tighter than previously existing bounds because they essentially scale with the magnitude of the true eigenvalue. For the case of rapidly decaying eigenvalues, these bounds correctly predict that the approximation error for small eigenvalues is much smaller than that of large eigenvalues.

3.1 Introduction

The kernel matrix is the square matrix obtained by evaluating the kernel function k on all pairs of object samples X_i, X_j. As the number of samples n tends to infinity, certain properties of the kernel matrix show a convergent behavior. In this chapter, we will focus on the eigenvalues of the kernel matrix. It is already known that these eigenvalues converge to the eigenvalues of the integral operator T_k with kernel function k with respect to the probability measure μ of the object samples X_i. One can therefore interpret the eigenvalues of the kernel matrix as statistical estimates of the eigenvalues of this integral operator.

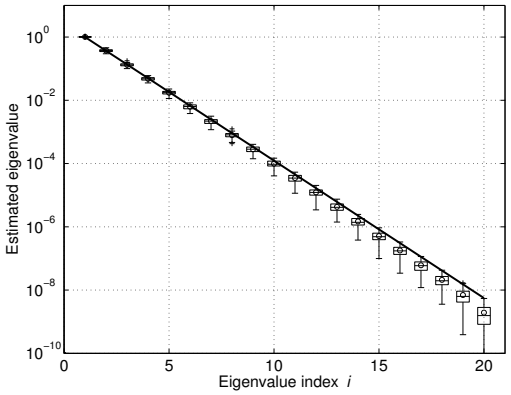
There exist many different ways to measure the difference between the estimated eigenvalues and the true eigenvalues, which are usually formalized as (at most countable infinite) point-sets in $\mathbb{C}$ or $\mathbb{R}$. Existing results have in common that the error between individual eigenvalues is measured on an absolute scale, independent of the magnitude of the true eigenvalue. Consequently, the predicted error for smaller eigenvalues is the same as that for larger eigenvalues.

However, numerical simulations suggest that this estimate is not realistic, but that smaller eigenvalues have much smaller fluctuations. Consider the following example: We construct a Mercer kernel using an orthogonal set of functions and a sequence of eigenvalues. To keep the example simple, consider Legendre polynomials $P_n(x)$ (Abramowitz and Stegun, 1972), which are orthogonal polynomials on $[-1, 1]$. We take the first 20 polynomials, and set $\lambda_i = \exp(-i)$. Then,

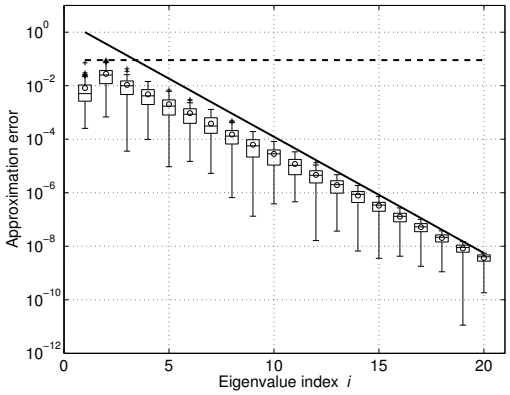
$$k(x, y) = \sum_{i=0}^{19} \nu_i e^{-i} P_i(x) P_i(y)$$

defines a Mercer kernel, where $\nu_i = 1/(2i + 1)$ are normalization factors such that P_i have unit norm with respect to the probability measure induced by $\mu([a, b]) = |b - a|/2$.

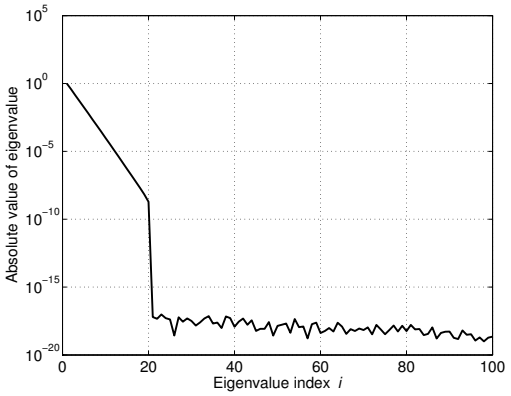
In Figure 3.1(a), the approximate eigenvalues of the kernel matrix constructed from 100 random samples in $[-1, 1]$ are plotted against the true eigenvalues. In Figure 3.1(b), the approximation errors (distance between approximate and true eigenvalue) are plotted. We see that the approximation error scales with the magnitude of the true eigenvalue such that the approximation error of smaller eigenvalues is much smaller than that of larger eigenvalues. In Figure 3.1(b), the smallest possible upper bound which does not scale with the magnitude of the true eigenvalues is plotted



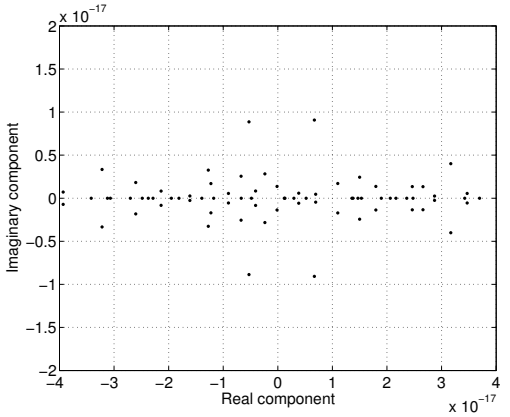
(a) True eigenvalues (solid line) and distributions of the approximate eigenvalues (box plots). These distributions have been estimated using 100 realizations of the 100 random points used to construct the kernel matrix.



(b) The approximation errors (box plots) and the true eigenvalues (solid line) for the largest 20 eigenvalues. The dashed line plots the smallest possible absolute bound on the approximation error as observed on the data.



(c) Mean of the absolute value of all 100 eigenvalues of the kernel matrix.



(d) A typical realization of the last 80 eigenvalues of the kernel matrix.

Figure 3.1: Approximated eigenvalues for kernel matrices with rapidly decaying eigenvalues have an approximation error which scales with the true eigenvalue. (For a discussion of the lower two figures, see Section 3.8.)

as a dashed line. We see that such a bound will fail to correctly reflect the fact that the approximation error scales with the magnitude of the true eigenvalue. For small eigenvalues, the absolute bound is overly pessimistic. A more accurate bound has to scale with the magnitude of the true eigenvalue. This observation is particularly important for the kernel functions employed in machine learning which typically have rapidly decaying eigenvalues.

We will derive a refinement of the convergence result which shows that the variance in the estimate depends on the magnitude of the true eigenvalue, such that estimates of smaller eigenvalues fluctuate much less than eigenvalues of larger eigenvalues. The resulting estimates of the approximation errors are consequently much tighter than previous results.

The relevance for machine learning is given by the fact that the kernel matrix is central to virtually all kernel methods. Knowledge of the eigenvalues of the kernel matrix can help to give insight into the workings of kernel methods. Implications of these results will be explored in later chapters.

The results in this chapter will be complemented by those of the next chapter, which study spectral projections and scalar products with eigenvectors of the kernel matrix. Both chapters combined result in a detailed analysis of the eigenstructure of the kernel matrix.

This chapter is structured as follows. Section 3.2 reviews the main results of this thesis in a less technical manner. Some definitions which are used throughout this chapter are introduced in Section 3.3. Section 3.4 reviews existing results on the convergence of eigenvalues. The notion of a relative-absolute bound is introduced in Section 3.5 together with a characterization of how a relative-absolute bound relates to uniform convergence of the eigenvalues. Some background information on classical perturbation theory for matrices in presented in Section 3.6. The basic relative-absolute perturbation bound in terms of the error matrices in developed in Section 3.7. Section 3.8 discusses the connection between relative-absolute bounds and finite precision arithmetics. We discuss two cases of kernel functions: in Section 3.9, kernels with bounded eigenfunctions, and in Section 3.10, bounded kernel functions. Section 3.11 discusses asymptotic rates of the bounds for different decay rates of eigenvalues. Some concrete examples are studied in Section 3.12. Finally, Section 3.13 discusses the results, while Section 3.14 concludes this chapter.

3.2 Summary of Main Results

We consider the eigenvalues of the *(normalized) kernel matrix* which is the square $n \times n$ matrix $\mathbf{K}_n$ with entries

$$[\mathbf{K}_n]_{ij} = \frac{1}{n} k(X_i, X_j), \tag{3.1}$$

where $X_1, \ldots, X_n$ are i.i.d. samples from the probability distribution μ on $\mathcal{X}$.

Convergence will be considered in terms of a *relative-absolute bound* on the error. If l_i is the ith approximate eigenvalue and λ_i the corresponding true eigenvalue, then the error is measured as

$$|l_i - \lambda_i| \leq \lambda_i C(r, n) + E(r, n). \tag{3.2}$$

The role of the r will be explained shortly. To understand why the introduction of a relative-absolute bound can improve the accuracy of the bound, consider an ordinary absolute bound first, $|l_i - \lambda_i| \leq E(n)$, where n is the sample size. The error term will be influenced most by the eigenvalues having the largest error. Experimental evidence suggests that these are the larger eigenvalues. Now if $E(n)$ were only to hold for the eigenvalues $\lambda_r, \ldots, \lambda_n$, the absolute bound could be much smaller. Fortunately, it turns out that for a finite number of eigenvalues, one can even construct a relative bound, so that $C(r, n)$ is a relative bound for the first r eigenvalues, while $E(r, n)$ bounds the error of the remaining eigenvalues. The resulting bound is much tighter than existing absolute bounds. These bounds decay as quickly as the eigenvalues until they reach a plateau at the size of $E(r, n)$. Therefore, one obtains truly relative bounds only for a finite number of eigenvalues while the remaining eigenvalues are subsumed under a (very small) absolute bound.

We will first derive a relative-absolute bound for the eigenvalues of the kernel matrix with error terms C and E given by the matrix norms of certain error matrices.

Theorem 3.3 (Basic Relative-Absolute Bound)
(Theorem 3.71 in the main text) The eigenvalues l_i of the kernel matrix converge to the true eigenvalues λ_i with a relative-absolute bound given by

$$|l_i - \lambda_i| \leq \lambda_i C(r,n) + E(r,n), \tag{3.4}$$

where

$$C(r,n) = \|\mathbf{C}_n^r\|, \qquad E(r,n) = \lambda_r + \|\mathbf{E}_n^r\|, \tag{3.5}$$

and

$$\mathbf{C}_n^r = {\mathbf{\Psi}_n^r}^\top \mathbf{\Psi}_n^r - \mathbf{I}_r, \qquad \mathbf{E}^r = \mathbf{K}_n - \mathbf{K}_n^{[r]}. \tag{3.6}$$

The columns of the matrix $\mathbf{\Psi}_n^r$ are given by the sample vectors of the eigenfunctions, and $\mathbf{E}_n^r$ measures the error of replacing the kernel function k with its r-degenerate approximation $k^{[r]}$ (see Section 2.4). This result is based on two well-known results for the perturbation of Hermitian matrices by Ostrowski and Weyl. The relative perturbation bound follows from Ostrowski's theorem for multiplicative perturbations by interpreting the kernel matrix as a multiplicative perturbation of a diagonal matrix containing the true eigenvalues by $\mathbf{\Psi}_n^r$. Since the eigenfunctions are orthogonal, the sample vectors will also become orthogonal asymptotically. The amount of non-orthogonality ${\mathbf{\Psi}_n^r}^\top \mathbf{\Psi}_n^r - \mathbf{I}_r$ controls the multiplicative perturbation of the eigenvalues. After that, the extension to all eigenvalues of the kernel matrix follows from an application of Weyl's theorem using the error matrix $\mathbf{E}_n^r$.

For matrices stored in real computers using finite precision arithmetics, it turns out that the characteristic shape of the relative-absolute bound accurately describes the eigenvalues of the kernel matrix. Due to the finite precision of real numbers stored in finite precision formats, the kernel matrix is stored with small perturbations. For example, for the ubiquitous `double` precision numbers, this perturbation is of the order of $\varepsilon = 10^{-16}$. Weyl's theorem applies in this case, stating that the resulting eigenvalues will be distorted on an absolute scale by the size of the perturbation. Therefore, actual eigenvalues will also decay quickly until they reach a plateau at around ε. Thus, although it might seem unsatisfactory from a theoretical point of view that the bound is not purely relative, from a practical point of view, the bound reflects the structure of actual eigenvalues very well.

The basic result depends on the norm of two error matrices. These are studied for two classes of kernel matrices. For both cases, detailed large deviation bounds for the size of the error terms are derived. These estimates result in a finite-sample size large deviation bound for the approximation error of the eigenvalues.

The first class of kernels is that of Mercer kernels whose eigenfunctions are uniformly bounded (for example a sine-basis).

Theorem 3.7 (Relative-Absolute Bound for Bounded Eigenfunctions)
(Theorem 3.94 in the main text) For Mercer kernels whose eigenfunctions are uniformly bounded by M, the eigenvalues converge with a relative-absolute bound with error terms

$$C(r,n) = O\left(n^{-\frac{1}{2}} r \sqrt{\log r}\right), \qquad E(r,n) = \lambda_r + M^2 \sum_{i=r+1}^{\infty} \lambda_i. \tag{3.8}$$

The slightly more intricate but also more interesting case is that of a uniformly bounded kernel function (more specifically, it suffices if the diagonal is uniformly bounded). An example for such kernel functions are radial-basis function kernels (rbf-kernels).

Theorem 3.9 (Relative-Absolute Bound for Bounded Kernel Functions)
(Theorem 3.135 and its immediate corollary in the main text) For Mercer kernels with a diagonal $x \mapsto k(x,x)$ which is uniformly bounded K, the eigenvalues converge with a relative-absolute bound

Class 1	polynomial decay	exponential decay
$C(r,n)$	$O\left(n^{-\frac{1}{2}} r\sqrt{\log r}\right)$	(same as for polynomial decay)
$E(r,n)$	$O\left(r^{1-\alpha}\right)$	$O\left(e^{-\beta r}\right)$
$r(n)$	$cn^{\frac{1}{2\alpha}}$	$\log cn^{\frac{1}{2\beta}}$
error rate	$O\left(n^{\frac{1-\alpha}{2\alpha}}\sqrt{\log n}\right)$	$O\left(n^{-\frac{1}{2}}(\log n)^{\frac{3}{2}}\right)$
Class 2	**polynomial decay**	**exponential decay**
$C(r,n)$	$O\left(r^{2+\frac{\alpha}{2}} n^{-\frac{1}{2}}\right)$	$O\left(e^{\frac{\beta}{2}r} r^2 n^{-\frac{1}{2}}\right)$
$E(r,n)$	$O\left(r^{1-\alpha} + r^{\frac{1-\alpha}{2}} n^{-\frac{1}{2}}\right)$	$O\left(e^{-\beta r} + e^{-\frac{\beta}{2}r} n^{-\frac{1}{2}}\right)$
$r(n)$	$cn^{\frac{1}{2+3\alpha}}$	$\log cn^{\frac{1}{3\beta}}$
error rate	$O\left(n^{\frac{1-\alpha}{2+3\alpha}}\right)$	$O\left(n^{-\frac{1}{3}}(\log n)^2\right)$

Figure 3.2: Asymptotic rates of C and E with respect to r and n for different classes of kernels and types of eigenvalue decay. Class 1 is the class of kernel functions with uniformly bounded eigenfunctions, while class 2 is the class of bounded Mercer kernels. The two types of decay are: polynomial decay $\lambda_i = O(i^{-\alpha})$ for $\alpha > 1$, and exponential decay $\lambda_i = O(e^{-\beta i})$ for $\beta > 0$. The term $C(r,n)$ denotes the relative error term, and $E(r,n)$ is the absolute error term. The table also lists minimal asymptotic rates $r(n) \to \infty$ as $n \to \infty$ such that $C(r(n),n) \to 0$ (for some $c > 0$), and the resulting overall error rate. For class 2, only the rate without the n^{-1} term is listed.

with error terms

$$C(r,n) = O\left(\lambda_r^{-\frac{1}{2}} r\sqrt{\log r} n^{-\frac{1}{2}} + \lambda_r^{-1} n^{-1} r \log r\right), \tag{3.10}$$

$$E(r,n) = \lambda_r + \Lambda_{>r} + O\left(\sqrt{\Lambda_{>r}} n^{-\frac{1}{2}} + n^{-1}\right), \tag{3.11}$$

where

$$\Lambda_{>r} = \sum_{i=r+1}^{\infty} \lambda_i. \tag{3.12}$$

Using an alternative large deviation bound, one obtains a bound without the n^{-1} terms:

$$C(r,n) = O\left(\lambda_r^{-\frac{1}{2}} r^2 n^{-\frac{1}{2}}\right), \tag{3.13}$$

$$E(r,n) = \Lambda_{>r} + O\left(\sqrt{\Lambda_{>r}} n^{-\frac{1}{2}}\right). \tag{3.14}$$

We consider two types of decay for eigenvalues, polynomial and exponential. We show that $\Lambda_{>r}$ obeys the following asymptotic rates:

$$\begin{aligned} &\text{polynomial:} && \lambda_i = O(i^{-\alpha}), (\alpha > 1), && \Lambda_{>r} = O(r^{1-\alpha}) \\ &\text{exponential:} && \lambda_i = O(e^{-\beta i}), (\beta > 0), && \Lambda_{>r} = O(e^{-\beta r}) \end{aligned}$$

Based on these examples, asymptotic rates with respect to r and n of $C(r,n)$ and $E(r,n)$ are derived. These are summarized in Table 3.2.

3.3 Preliminaries

We consider Mercer kernels as introduced in Chapter 2. Let μ be the common distribution of the $X_i \in \mathcal{X}$, $(\lambda_i)_{i\in\mathbb{N}}$ a sequence of non-negative real numbers in ℓ^1, and $(\psi_i)_{i\in\mathbb{N}}$ a family of orthonormal functions in $\mathcal{H}_\mu(\mathcal{X})$. Then, the following formula defines a Mercer kernel: for all $x, y \in \mathcal{X}$, let

$$k(x,y) = \sum_{i=1}^{\infty} \lambda_i \psi_i(x)\psi_i(y). \tag{3.15}$$

As discussed in Chapter 2, the λ_i are the eigenvalues of the integral operator T_k, which is defined on $\mathcal{H}_\mu(\mathcal{X})$ by

$$\mathcal{H}_\mu(\mathcal{X}) \ni f \longmapsto T_k f := \left(\mathcal{X} \ni x \mapsto \int_{\mathcal{X}} k(x,y) f(y) \mu(dy) \right) \in \mathcal{H}_\mu(\mathcal{X}), \tag{3.16}$$

and ψ_i are its eigenfunctions.

Throughout this thesis we assume that eigenvalues are repeated according to their geometric multiplicity and that eigenvalues and eigenvectors have been ordered such that the eigenvalues are in non-increasing order.

The (normalized) kernel matrix is the $n \times n$ matrix $\mathbf{K}_n$ with entries

$$[\mathbf{K}_n]_{ij} = \frac{1}{n} k(X_i, X_j). \tag{3.17}$$

Recall that the set of all eigenvalues of $\mathbf{K}_n$ is denoted by $\lambda(\mathbf{K}_n)$, whereas $\lambda_i(\mathbf{K}_n)$ is the ith eigenvalue of $\mathbf{K}_n$ sorted in non-increasing order.

We will later use degenerate kernels to approximate a given kernel function. The approximation is obtained by truncating the sum in (3.15) to a finite number of terms. We write

$$k^{[r]}(x,y) = \sum_{i=1}^{r} \lambda_i \psi_i(x) \psi_i(y) \tag{3.18}$$

for the kernel function truncated to the first r terms. The kernel matrix for $k^{[r]}$ is denoted by $\mathbf{K}_n^{[r]}$. The r-error function measures the truncation error:

$$e^r(x,y) = k(x,y) - k^{[r]}(x,y) = \sum_{i=r+1}^{\infty} \lambda_i \psi_i(x) \psi_i(y). \tag{3.19}$$

The *truncation error matrix* is

$$\mathbf{E}_n^r = \mathbf{K}_n - \mathbf{K}_n^{[r]}. \tag{3.20}$$

3.4 Existing Results on the Eigenvalues

Our goal is a bound on the approximation error which scales with the eigenvalues. Such a bound will lead to much better estimates if the eigenvalues decay rapidly. However, it is in general hard to obtain a priori information on the decay rate of algorithms for the settings occurring in machine learning.

The reason is that rigorous results on the rate of decay of eigenvalues are available only for certain special cases, for example for integral operators with respect to uniform measures on hypercubes. Two examples are results by Hille-Tamarkin (see (Engl, 1997, Theorem 8.3)) or Chang (see Weyl (1968)) which show that the rate of decay is linked to the smoothness of the kernel, to the effect that eigenvalues of smoother kernels decay faster. However, in machine learning, the underlying measure with respect to which the integral operator is defined is not uniform but depends on the learning problem at hand. Since this probability measure has a strong effect on the eigenvalues, these classical results are not readily applicable.

In Section 3.11, we will discuss exemplary cases assuming that the eigenvalues decay polynomially or exponentially.

Let us review some results on the asymptotic behavior of the eigenvalues of the kernel matrix. We consider Mercer kernels k, which are symmetric kernels which generate positive semi-definite self-adjoint operators. Furthermore, the eigenvalues are summable which implies that the eigenvalues are also square summable, such that the kernels are also Hilbert-Schmidt kernels.

Now as state above, it is well-known that the eigenvalues of the (normalized) kernel matrix converge to the eigenvalues of the associated integral operator T_k. Convergence of $\mathbf{K}_n$ to T_k follows from general principles because one can show that the operator

$$\tilde{\mathbf{K}}_n f(x) = \frac{1}{n} \sum_{i=1}^{n} k(x, X_i) f(X_i). \tag{3.21}$$

has the same eigenvalues as the matrix $\mathbf{K}_n$ and approximates T_k in an appropriate sense by Monte Carlo integration. (This fact is easy to see for fixed x. Some effort has to be put into proving this on some appropriate function space). Therefore, for large n, $\mathbf{K}_n$ can be thought of as a small perturbation of T_k and therefore approximates the eigenvalues of T_k.

This approach is for example taken by von Luxburg (2004). Using this functional analytic approach, it is possible to derive approximation results which hold for fairly general cases, also for operators which are not self-adjoint. The power of this approach is at the same time a possible shortcoming, because the resulting error bounds are not very specific. For example, the approximation error is usually measured as the distance of one approximate eigenvalue to the point set of true eigenvalues. More cannot be said for general operators, but for self-adjoint positive operators, more specific results are obtainable.

In Koltchinskii and Giné (2000), a similar approach is taken with the significant modification that the comparison is performed on $\mathbb{R}^n$: instead of embedding $\mathbf{K}_n$ into some function space, a finite-dimensional image of T_k is computed. In order to compare the finitely many eigenvalues of $\mathbf{K}_n$ with the infinite sequence of eigenvalues of T_k, some procedure has to be constructed.

In the above paper, the convergence result is stated with respect to the following metric. We assume that the eigenvalues are all non-negative and sorted in non-increasing order, repeated according to their multiplicity (which means that an eigenvalue whose eigenspace has dimensions d is repeated d times). Thus, we obtain eigenvalue tuples and sequences

$$\lambda(\mathbf{K}_n) = (l_1, \ldots, l_n), \qquad l_1 \geq \ldots \geq l_n \tag{3.22}$$

$$\lambda(T_k) = (\lambda_1, \lambda_2, \ldots), \qquad \lambda_1 \geq \lambda_2 \geq \ldots. \tag{3.23}$$

To compare the eigenvalues, $\lambda(\mathbf{K}_n)$ is first embedded into ℓ^1 by filling up the n-vector by zeros,

$$\lambda(\mathbf{K}_n) = (l_1, \ldots, l_n, 0, 0, \ldots). \tag{3.24}$$

Then, the distance between these (countably) infinite sequence is defined as

$$\delta_2(\lambda(\mathbf{K}_n), \lambda(T_k)) = \inf_{\pi \in \mathfrak{S}(\mathbb{N})} \sum_{i=1}^{\infty} \left(l_i - \lambda_{\pi(i)}\right)^2, \tag{3.25}$$

where $\mathfrak{S}(\mathbb{N})$ is the set of all bijections on $\mathbb{N}$. With these definitions, one can state the following theorem:

Theorem 3.26 *(Koltchinskii and Giné, 2000, Theorem 3.1)*
Let k be a Mercer kernel, then

$$\delta_2(\lambda(\mathbf{K}_n), \lambda(T_k)) \to_{a.s.} 0. \tag{3.27}$$

Actually, the theorem is proven for the matrix $\mathbf{K}_n$ with the diagonal elements set to zero, but for the Mercer kernels we consider, the same result holds.

Sometimes, some standard reference on the numerical analysis of integral equation is cited (for example (Baker, 1977; Anselone, 1971; Atkinson, 1997)) for results on the convergence of eigenvalues. Unfortunately, these results are in general not applicable to the case one is interested in for a machine learning context. For the numerical analysis of integral equations, one usually considers integral operators defined on compact domains (for example closed intervals and products of those), and T_k is approximated by some classical quadrature method. In contrast, in machine learning, the support of the probability measure μ is in general not compact (for example if μ is a mixture of Gaussians), and the integration is approximated by means of Monte Carlo integration as in (3.21). This does not

mean that convergence does not take place, only that the proofs cannot directly be transferred to the machine learning setting (also compare the discussion in von Luxburg (2004)). Works like the one by Koltchinskii and Giné (2000) which approach the question from a probabilistic perspective treat the case which is relevant for machine learning contexts.

Next we address the convergence speed. The paper Koltchinskii and Giné (2000) contains estimates for the convergence in the δ_2-metric, but we are actually more interested in the behavior of single eigenvalues which is treated by central limits theorem in that work.

More specifically, in (Koltchinskii and Giné, 2000, Theorem 5.1), a central limit theorem type result is derived for the finite-dimensional distributions of the approximation errors

$$\text{error} = \sqrt{n}(\lambda(\mathbf{K}_n) - \lambda(T_k)), \tag{3.28}$$

more specifically, the asymptotic distribution of the approximation errors of finite subsets of eigenvalues is considered.

The limit distribution is stated with respect to a Gaussian process on $\mathcal{H}_\mu(\mathcal{X})$. This Gaussian process G is a family of random variables indexed by functions $f \in \mathcal{H}_\mu(\mathcal{X})$. For each fixed f, G_f is a Gaussian random variable with mean zero. The covariance between G_f and G_g is defined as

$$\mathbf{Cov}(G_f, G_g) = \mathbf{E}(G_f G_g) = \mathbf{E}_\mu(fg) - \mathbf{E}_\mu(f)\mathbf{E}_\mu(g) = \mathbf{Cov}(f, g). \tag{3.29}$$

In other words, G is a centered Gaussian process which has the covariance structure of the underlying index space: $\mathbf{Cov}(G_f, G_g) = \mathbf{Cov}(f, g)$.

With these definitions, Theorem 5.1 in Koltchinskii and Giné (2000) states that under certain regularity conditions on k, the finite-dimensional distributions of (3.28) converge to those of

$$\bigoplus_{j=1}^{\infty} \lambda(\mathbf{\Gamma}_{i_j}), \tag{3.30}$$

where $\oplus$ denotes concatenation of vectors, $(\lambda_{i_j})_j$ is the subsequence of unique eigenvalues with the convention that i_j is the first occurrence of the eigenvalue λ_{i_j}, and $\mathbf{\Gamma}_{i_j}$ are the (random) matrices

$$\mathbf{\Gamma}_{i_j} = \left(\lambda_{i_j} G_{\psi_p \psi_q}\right)_{p,q=i_j}^{i_{j+1}-1}. \tag{3.31}$$

Thus, each eigenvalue corresponds to the eigenvalues of the matrix $\mathbf{\Gamma}_{i_j}$ having Gaussian entries, whose size is given by the multiplicity of the eigenvalue.

Let us assume for a moment that each eigenvalue has multiplicity 1. Then, the subsequence (i_j) is just $\mathrm{id}_{\mathbb{N}}$. Furthermore, $\mathbf{\Gamma}_j = \lambda_j G_{\psi_j^2}$. Since the matrices are then scalar values, the eigenvalues are just the single entry, and the limiting distribution simplifies greatly. The finite-dimensional distributions of (3.28) converge to the finite-dimensional distributions of the random sequence

$$(\lambda_1 G_{\psi_1^2}, \lambda_2 G_{\psi_2^2}, \ldots). \tag{3.32}$$

Finally, if we just consider one eigenvalue l_i, we get that

$$\sqrt{n}(l_i - \lambda_i) \rightsquigarrow \lambda_i G_{\psi_i^2}, \tag{3.33}$$

and the variance depends on the fourth moment of the eigenfunction ψ_i since $\mathbf{Var}(G_{\psi_i^2}) = \mathbf{Var}(\psi_i^2)$. Therefore, the central limit theorem result already confirms our experimental observation that the variance of the estimated eigenvalues scales with the magnitude of the true eigenvalue (although we have not yet considered the variance of ψ_i^2).

Now in the theoretical analysis of machine learning algorithms, the usefulness of central limit theorems is somewhat limited mainly due to two reasons. First of all, a central limit theorem is an asymptotic result. Although experimental experience tells us that the normal approximations are usually quite

reliable, there is no statement on the speed of convergence, such that one does not know how many sample points are enough to achieve a certain error. Second of all, the processes involved in machine learning are often too complex to allow for the computation of properties as fundamental as even the mean or the variance in an exact fashion. Therefore, one often only bounds the error given a certain confidence. This kind of information can then be used for further computations. We are thus interested in finite-size confidence bounds. These have the following form. Given a confidence $0 < \delta < 1$ we want to obtain an estimate C such that with probability larger than $1 - \delta$,

$$|l_i - \lambda_i| \leq C. \tag{3.34}$$

In some way, this forms a reduction of the information about the distribution of $l_i - \lambda_i$ to a single number which hopefully captures the essence of the behavior of l_i. This single number can then be used in more discrete, computer science type derivations.

Such finite-sample size confidence bounds have been studied in (Shawe-Taylor et al., 2002a,b; Shawe-Taylor and Williams, 2003; Zwald et al., 2004). The main idea lies in transforming the algebraic eigenvalue problem into an optimization problem over a random function. The starting point is a variational characterization of the eigenvalues attributed to Courant and Fischer:

Theorem 3.35 (Courant-Fischer) *(see for example, Kato (1976)) Let* $\mathbf{A}$ *be a Hermitian* $n \times n$ *matrix. Then,*

$$\lambda_i(\mathbf{A}) = \max_{\substack{V \subset \mathbb{R}^n, \\ \dim(V)=i}} \min_{\substack{v \in V, \\ \|v\|=1}} v^\top \mathbf{A} v. \tag{3.36}$$

The eigenvalues of the kernel matrix thus appear as the solution of an optimization problem. The question of how much the approximate eigenvalues fluctuate is reduced to the question how much the objective function fluctuates and how that influences the solution of the optimization problem. Using McDiarmid's inequality (McDiarmid, 1989), one can show that with increasing sample size, the solution concentrates around its expectation, such that the eigenvalues become concentrated as well:

Theorem 3.37 *(Shawe-Taylor et al., 2002b, Theorem 4)*
Let k *be a Mercer kernel on* $\mathcal{X}$ *and* $|k(x, y)| \leq K$ *for all* $x, y \in \mathcal{X}$*. Then, for all* $n \in \mathbb{N}$ *and* $\varepsilon > 0$*,*

$$\mathbf{P}\left\{|l_i - \mathbf{E}l_i| \geq \varepsilon\right\} \leq 2 \exp\left(-\frac{2\varepsilon^2 n}{K^2}\right). \tag{3.38}$$

One can extend the variational characterization and the result to sums of leading eigenvalues and sums of all but the first few eigenvalues.

In Zwald et al. (2004), a more refined analysis of the case of general Mercer kernels is undertaken based on some recent concentration results.

Theorem 3.39 *(Zwald et al., 2004, Theorem 4) Let* k *be a Mercer kernel on* $\mathcal{X}$ *with* $|k(x, y)| \leq K$ *for all* $x, y \in \mathcal{X}$*. Then, for all* $n \in \mathbb{N}$ *with probability larger than* $1 - 3e^{-x}$*,*

$$-K\sqrt{\frac{x}{2n}} \leq \sum_{i=1}^{r} l_i - \sum_{i=1}^{r} \lambda_i \leq 2\sqrt{\frac{r}{n}\kappa^2} + 3M\sqrt{\frac{x}{2n}}, \tag{3.40}$$

with $\kappa^2 = \frac{1}{n}\sum_{i=1}^{n} k^2(X_i, X_i) \leq K^2$*.*

Both works derive the sort of finite-sample size confidence bounds which we will be looking for, but note that the confidence does not depend on the magnitude of the true eigenvalue, and is in fact the same for all eigenvalues. Therefore, these bounds do not reflect the true behavior of the approximated eigenvalues. Above we stated the hope that reducing the whole distribution of l_i to a single number using a confidence bound, we still grasp the essential properties of the distribution. Here we see that these bounds fail at this requirement, because by (3.33), the variance asymptotically depends on the eigenvalue, but in (3.38) and (3.40), the bound is the same for all sets of eigenvalues. In this chapter, we will try to derive a bound which reflects the true behavior of the approximate eigenvalues better.

3.5 Relative-Absolute Error Bounds

We consider pair-wise bounds between the ith approximate eigenvalue l_i and the ith true eigenvalue λ_i, both with respect to a non-increasing ordering. We are interested in obtaining a tight bound for $|l_i - \lambda_i|$. An absolute bound has the form

$$|l_i - \lambda_i| \leq E(n), \tag{3.41}$$

giving a uniform estimate of the error. Since $E(n)$ has to be an upper bound for every i, this means that $E(n)$ will be determined to a large extend by the large errors and the error estimate will be unrealistically large for the remaining eigenvalues. This observation is in particular true if the λ_i decay quickly.

We study a refinement of absolute bounds which are obtained by introducing a relative term (for $1 \leq i \leq n$ and for all $r, n \in \mathbb{N}$)

$$|l_i - \lambda_i| \leq \lambda_i C(r,n) + E(r,n), \tag{3.42}$$

where $C, E\colon \mathbb{N} \times \mathbb{N} \to \mathbb{R}_{\geq 0}$. The term $C(r,n)$ will be called the *relative error term*, while $E(r,n)$ will be called the *absolute error term*. Note that one actually obtains a family of upper bounds depending on the *truncation point* r. The idea is that the relative error $C(r,n)$ takes care of the approximation error for the first r eigenvalues, while $E(r,n)$ can then reflect the approximation error for the eigenvalues $\lambda_{r+1}, \ldots, \lambda_n$. Therefore, the overall bound will be much tighter.

In any case, for each fixed r, the bound (3.42) is a valid upper bound to the approximation error. The smallest upper bound can be derived from (3.42) by considering for each fixed i the minimum over all upper bounds with r varying from 1 to n:

$$|\lambda_i(\mathbf{K}_n) - \lambda_i| \leq \min_{1 \leq r \leq n} \left(\lambda_i C(r,n) + E(r,n)\right). \tag{3.43}$$

In general, there is no such thing as an optimal choice of r with regard to all eigenvalues $\lambda_1, \ldots, \lambda_n$ at once, but the choice has to depend on i. We will see later that the bound will depend on a number of parameters. Therefore, it is in general also not possible to determine the optimal choice of $r(i)$ in a nice closed form.

We close with the following lemma on the convergence of the approximate eigenvalues to the true eigenvalues in terms of relative-absolute bounds.

Lemma 3.44 *Let $L_n = (l_{n,1}, \ldots, l_{n,n})$, and $\Lambda = (\lambda_i)_{i=1}^{\infty}$, both sorted in non-increasing order. Let a relative-absolute bound as in* (3.42) *hold. Convergence of L_n to Λ with respect to the measure*

$$d(L_n, \Lambda) = \max_{1 \leq i \leq n} |l_{n,i} - \lambda_i|, \tag{3.45}$$

is implied by

$$\forall r \in \mathbb{N}\colon \lim_{n \to \infty} C(r,n) = 0, \qquad \lim_{r \to \infty} \lim_{n \to \infty} E(r,n) = 0. \tag{3.46}$$

Proof By Equation (3.42),

$$\begin{aligned} d(L_n, \Lambda) &= \max_{1 \leq i \leq n} |l_{n,i} - \lambda_i| \\ &\leq \max_{1 \leq i \leq n} \lambda_i C(r,n) + E(r,n) \\ &\leq \lambda_1 C(r,n) + E(r,n), \end{aligned}$$

because the λ_i are sorted in non-increasing order. This inequality holds for all r, n, therefore:

$$\begin{aligned} \lim_{n \to \infty} d(L_n, \Lambda) &= \lim_{r \to \infty} \lim_{n \to \infty} d(L_n, \Lambda) \\ &\leq \lim_{r \to \infty} \lim_{n \to \infty} \left(\lambda_1 C(r,n) + E(r,n)\right) \\ &= \lim_{r \to \infty} \lim_{n \to \infty} E(r,n) = 0, \end{aligned}$$

by the conditions on C and E. ■

Later in this chapter, we will see that for degenerate kernels, it is even possible to derive a pure relative bound. For general kernels, one obtains a relative-absolute bound.

3.6 Perturbation of Hermitian Matrices

The relative-absolute perturbation bounds rely on two classical result from the theory of perturbation of Hermitian matrices (complex matrices for which $\mathbf{A}^* = \mathbf{A}$). This theory considers the question how the eigenvalues of a matrix $\mathbf{A} \in \mathbb{M}_n$ change when $\mathbf{A}$ is perturbed, for example, by adding a matrix $\mathbf{E}$ which has a small norm compared to $\mathbf{A}$, resulting in $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{E}$. We list several examples, the last two of which will be used later on.

For general matrices, the Bauer-Fike Theorem tells us that we can bound the perturbation of an individual eigenvalue from the set of the eigenvalues of $\mathbf{A}$: Let $\tilde{\lambda}$ be an eigenvalue of $\tilde{\mathbf{A}}$. Then,

$$\min_{\lambda \in \lambda(\mathbf{A})} |\lambda - \tilde{\lambda}| \leq \kappa(\mathbf{X}) \|\mathbf{E}\|, \tag{3.47}$$

where $\mathbf{X}$ is the matrix whose columns are the eigenvectors of $\mathbf{A}$, and $\kappa(\mathbf{X})$ denotes the condition of $\mathbf{X}$. The optimal condition is given if the eigenvectors are orthogonal (for example, if $\mathbf{A}$ is Hermitian). Then, $\kappa(\mathbf{X}) = 1$, and the deviation of the eigenvalues is bounded by the norm of $\mathbf{E}$.

The more is known about the structure of the matrix, or the more restricted the class of matrices is which are considered, the more specific the bound becomes. For example, if $\mathbf{A}$ and $\tilde{\mathbf{A}}$ are both Hermitian, one can compute distances between the true and the perturbed eigenvalues for pairs of eigenvalues. Let $\lambda_1 \geq \ldots \geq \lambda_n$ be the eigenvalues of $\mathbf{A}$, and $\tilde{\lambda}_1 \geq \ldots \geq \tilde{\lambda}_n$ be those of $\tilde{\mathbf{A}}$, then the Hoffman-Wielandt theorem (Hoffman and Wielandt, 1953) states that

$$\sum_{i=1}^{n} (\tilde{\lambda}_i - \lambda_i)^2 \leq \|\mathbf{E}\|_F^2, \tag{3.48}$$

where $\|\mathbf{E}\|_F$ is the Frobenius norm of $\mathbf{E}$ (see page 13).

The pairing between the eigenvalues of $\mathbf{A}$ and $\tilde{\mathbf{A}}$ ensures that each individual eigenvalue is approximated well. A different way to state this is the following theorem, which will be used in the derivation of the relative-absolute perturbation bound later on:

Theorem 3.49 (Weyl) *(Horn and Johnson, 1985, Theorem 4.3.1)*
Let $\mathbf{A}, \mathbf{E}$ be Hermitian $n \times n$ matrices. Then, for each $1 \leq i \leq n$,

$$\lambda_i(\mathbf{A}) + \lambda_n(\mathbf{E}) \leq \lambda_i(\mathbf{A} + \mathbf{E}) \leq \lambda_i(\mathbf{A}) + \lambda_1(\mathbf{E}), \tag{3.50}$$

which implies that

$$|\lambda_i(\mathbf{A}) - \lambda_i(\mathbf{A} + \mathbf{E})| \leq \|\mathbf{E}\|. \tag{3.51}$$

The main difference between Weyl's theorem and the Hoffman-Wielandt theorem is that Weyl's theorem is a bit more specific with respect to how large the deviation for a single pair $(\tilde{\lambda}_i, \lambda_i)$ is, whereas the Hoffman-Wielandt states that that the overall deviation is small. For example, it could be the case that all of the measured approximation error is contained in the smallest eigenvalue pair only. Since we want to specifically exclude this situation, we will use Weyl's theorem.

A different type of perturbation is given by mapping a Hermitian matrix $\mathbf{A}$ to $\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A}\mathbf{S}^*$. The size of the magnitude has to be measured differently in this case, though. Note that if $\mathbf{S}^*\mathbf{S} = \mathbf{I}$, then conjugating with $\mathbf{S}$ amounts to an orthogonal change of basis, which does not change the eigenvalues at all. Therefore, the amount of perturbation will be measured by the non-orthogonality of $\mathbf{S}$. Sylvester's Law of Inertia tells us that the sign of eigenvalues is not changed by this operation. The following perturbation result can thus be understood as a quantitative variant of the Sylvester's Law.

Theorem 3.52 (Ostrowski)
(Horn and Johnson, 1985, Theorem 4.5.9, Corollary 4.5.11)
Let $\mathbf{A}$ *be a Hermitian* $n \times n$ *matrix, and* $\mathbf{S}$ *an* $n \times n$ *matrix. Then, for each* $1 \leq i \leq n$*, there exists an nonnegative real* θ_i *with* $\lambda_n(\mathbf{SS}^*) \leq \theta_i \leq \lambda_1(\mathbf{SS}^*)$*, such that*

$$\lambda_i(\mathbf{SAS}^*) = \theta_i \lambda_i(\mathbf{A}). \tag{3.53}$$

For our purposes, we need a reformulation of Ostrowski's theorem and an extension to the case of non-square perturbation matrices $\mathbf{S}$.

Corollary 3.54 *Under the conditions of Ostrowski's theorem, it holds that*

$$|\lambda_i(\mathbf{SAS}^*) - \lambda_i(\mathbf{A})| \leq |\lambda_i(\mathbf{A})| \|\mathbf{S}^*\mathbf{S} - \mathbf{I}\|. \tag{3.55}$$

Proof Let $\mathbf{Q} = \mathbf{S}^*\mathbf{S}$, $\lambda_i = \lambda_i(\mathbf{A})$, and $\tilde{\lambda}_i = \lambda_i(\mathbf{SAS}^*)$. Then, by (3.53),

$$\tilde{\lambda}_i \leq \lambda_1(\mathbf{Q})\lambda_i \quad \Rightarrow \quad \tilde{\lambda}_i - \lambda_i \leq \lambda_i(\lambda_1(\mathbf{Q}) - 1), \tag{3.56}$$

and analogously,

$$\tilde{\lambda}_i - \lambda_i \geq \lambda_i(\lambda_n(\mathbf{Q}) - 1). \tag{3.57}$$

Combining both gives

$$\begin{aligned} |\tilde{\lambda}_i - \lambda_i| &\leq \max(-\lambda_i(\lambda_n(\mathbf{Q}) - 1), \lambda_i(\lambda_1(\mathbf{Q}) - 1)) \\ &\leq |\lambda_i| \max(|\,\lambda_n(\mathbf{Q}) - 1|, |\,\lambda_1(\mathbf{Q}) - 1|) \\ &\leq |\lambda_i| \max_{i \in \{1,\ldots,n\}} |\,\lambda_i(\mathbf{Q}) - 1| \\ &= |\lambda_i| \|\mathbf{Q} - \mathbf{I}\|, \end{aligned} \tag{3.58}$$

because $\mathbf{Q} - \mathbf{I}$ is Hermitian. ■

Corollary 3.59 *Ostrowski's theorem and its corollary also hold under the condition that* $\mathbf{A}$ *is a Hermitian* $r \times r$ *matrix and* $\mathbf{S}$ *a general* $n \times r$ *matrix.*

Proof (i) If $n > r$, we extend $\mathbf{A}$ and $\mathbf{S}$ to $n \times n$ matrices $\mathbf{A}'$, respectively $\mathbf{S}'$:

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A} & 0 \\ 0 & 0 \end{bmatrix}, \qquad \mathbf{S}' = \begin{bmatrix} \mathbf{S} & 0 \end{bmatrix}. \tag{3.60}$$

We have to check how the eigenvalues of the extended matrices relate to those of the original matrices for $\mathbf{SAS}^*$, $\mathbf{A}$, and $\mathbf{SS}^*$.

With these definitions, since $\mathbf{A}$ is block-diagonal, $\lambda(\mathbf{A}') = \lambda(\mathbf{A}) \cup \lambda(\mathbf{0}_{n-r})$, and $\mathbf{A}'$ has the same eigenvalues as $\mathbf{A}$, except for an added multiplicity of $r - n$ for the eigenvalue 0.

Let us show that $\mathbf{S}'\mathbf{A}'\mathbf{S}'^* = \mathbf{SAS}^*$: Denote the entries of matrices by their respective lowercase letters, then for all $1 \leq i, j \leq n$,

$$[\mathbf{S}'\mathbf{A}'\mathbf{S}'^*]_{ij} = \sum_{k,\ell=1}^{n} s'_{ik} a'_{k\ell} \overline{s'_{j\ell}} = \sum_{k,\ell=1}^{r} s'_{ik} a'_{k\ell} \overline{s'_{j\ell}}, \tag{3.61}$$

because $a'_{k\ell} = 0$ if $k > r$ or $\ell > r$. But for $1 \leq k, \ell \leq r$, $s'_{ik} = s_{ik}$ and $a'_{ik} = a_{ik}$. Therefore, the last display equals

$$\sum_{k,\ell=1}^{r} s'_{ik} a'_{k\ell} \overline{s'_{j\ell}} = \sum_{k,\ell=1}^{r} s_{ik} a_{k\ell} \overline{s_{j\ell}} = [\mathbf{SAS}^*]_{ij}. \tag{3.62}$$

Consequently, $\lambda_i(\mathbf{S}'\mathbf{A}'\mathbf{S}'^*) = \lambda_i(\mathbf{SAS}^*)$ for all $1 \leq i \leq n$.

Finally, one can compute that $\mathbf{S}'\mathbf{S}'^* = \mathbf{S}\mathbf{S}^*$, such that the eigenvalues are equal. This completes the case $n > r$.

(ii) Now if $n < r$, extend the $n \times r$ matrix $\mathbf{S}$ to the $r \times r$ matrix $\mathbf{S}'$ by setting

$$\mathbf{S}' = \begin{bmatrix} \mathbf{S} \\ 0 \end{bmatrix}. \tag{3.63}$$

We compute

$$[\mathbf{S}'\mathbf{A}\mathbf{S}'^*]_{ij} = \sum_{k,\ell=1}^{r} s'_{ik} a_{k\ell} \overline{s'_{j\ell}} = \begin{cases} [\mathbf{S}\mathbf{A}\mathbf{S}^*]_{ij} & 1 \leq i,j \leq n \\ 0 & n < i \leq r \text{ or } n < j \leq r, \end{cases} \tag{3.64}$$

because $s'_{ik} = s_{ik}$ for $1 \leq i \leq n$ and $s'_{ik} = 0$ for $i > n$. Therefore, $\mathbf{S}'\mathbf{A}\mathbf{S}'^*$ is block-diagonal, and $\lambda(\mathbf{S}'\mathbf{A}\mathbf{S}'^*) = \lambda(\mathbf{S}\mathbf{A}\mathbf{S}^*) \cup \lambda(\mathbf{0}_{r-n})$. For $\mathbf{A} = \mathbf{I}$, it follows that $\lambda(\mathbf{S}'\mathbf{S}'^*) = \lambda(\mathbf{S}\mathbf{S}^*) \cup \lambda(\mathbf{0}_{r-n})$, but note that since $n < r$, $\mathbf{S}\mathbf{S}^*$ is singular anyway, therefore, the smallest singular value of $\mathbf{S}\mathbf{S}^*$ is also zero, as that of $\mathbf{S}'\mathbf{S}'^*$.

In other words, if $\mathbf{S}$ is non-square, we can extend $\mathbf{S}$ and $\mathbf{A}$ such that $\mathbf{S}$ and $\mathbf{A}$ are square matrices of the same dimension, and only the multiplicity of the null eigenvalue is increased. The remaining eigenvalues are as in the original case. The bounds from Ostrowski's theorem for the extended matrices thus also hold for the original case. ■

3.7 The Basic Relative-Absolute Perturbation Bound

In this section, we derive the basic relative-absolute perturbation bound. This result is the main result of this chapter, in the sense that the remaining sections treat two specific classes of kernel functions and derive estimates for the size of two error matrices on which the relative-absolute perturbation bound is based. The improvement in the tightness of the bound compared to absolute bounds is a consequence of the bound derived in this section, not of the estimates of the next sections.

We first begin with the case where k is degenerate. In this case we can actually derive a pure relative bound. This result will be stated for the case of n arbitrary points $x_1, \ldots, x_n$. Note that, strictly speaking, since elements of $\mathcal{H}_\mu(\mathcal{X})$ are equivalence classes of functions which differ only on a set of measure zero, point evaluations are not well-defined. However, we will later on only consider the case where the points are i.i.d. samples distributed as μ. This case is well-defined, since only the resulting distributions are relevant.

Theorem 3.65 *Let k be an r-degenerate Mercer kernel on the Hilbert space $\mathcal{H}_\mu(\mathcal{X})$ with eigenvalues $\lambda_1, \ldots, \lambda_r$ and eigenfunctions $\psi_1, \ldots, \psi_r$. Furthermore, let $\mathbf{K}_n$ be the normalized kernel matrix induced by n points $x_1, \ldots, x_n \in \mathcal{X}$. Then, it holds that for all $1 \leq i \leq r$,*

$$|\lambda_i(\mathbf{K}_n) - \lambda_i| \leq \lambda_i \|\mathbf{\Psi}_n^{r\top} \mathbf{\Psi}_n^r - \mathbf{I}_r\|, \tag{3.66}$$

where $\mathbf{\Psi}_n^r$ is the $n \times r$ matrix with entries

$$[\mathbf{\Psi}_n^r]_{i\ell} = \frac{1}{\sqrt{n}} \psi_\ell(x_i). \tag{3.67}$$

Proof The key step consists in using Equation (3.15) to rewrite the normalized kernel matrix $\mathbf{K}_n$ such that Ostrowski's Theorem can be applied[1].

From (3.15), the ijth entry k_{ij} of $\mathbf{K}_n$ is

$$k_{ij} = \frac{1}{n} \sum_{\ell=1}^{r} \lambda_\ell \psi_\ell(x_i) \psi_\ell(x_j). \tag{3.68}$$

[1] A similar construct was used by Koltchinskii and Giné (2000), but in conjunction with a bound which measures the absolute error.

This expression can be written in matrix notation using the following definition: Let $\mathbf{\Psi}_n^r$ be as in the statement of the theorem, and let $\mathbf{\Lambda}_r = \operatorname{diag}(\lambda_1, \ldots, \lambda_r)$. Then,

$$[\mathbf{\Psi}_n^r \mathbf{\Lambda}_r \mathbf{\Psi}_n^{r\top}]_{ij} = \frac{1}{n} \sum_{k,\ell=1}^{r} \psi_k(x_i)[\mathbf{\Lambda}_r]_{k\ell}\psi_\ell(x_j) = \frac{1}{n}\sum_{\ell=1}^{r} \psi_\ell(x_i)\lambda_\ell\psi_\ell(x_j) = k_{ij}, \tag{3.69}$$

because $\mathbf{\Lambda}_r$ is a diagonal matrix.

Therefore, we want to compare the eigenvalues of $\mathbf{K}_n$ to those of T_k, $\lambda_1, \ldots, \lambda_r$, but trivially, these are also the eigenvalues of $\mathbf{\Lambda}_r$. Therefore, we can equivalently compare the eigenvalues of $\mathbf{\Lambda}_r$ and $\mathbf{\Psi}_n^r \mathbf{\Lambda}_r \mathbf{\Psi}_n^{r\top}$. But this is exactly the situation which is addressed by Ostrowski's theorem. Apply Corollary 3.59 and (3.66) follows. ■

Next, we extend the case to general Mercer kernels by reducing the general case to the basic case as follows. Recall that $\mathbf{K}_n^{[r]}$ is the normalized kernel matrix of the truncated kernel function $k^{[r]}$. For n fixed points $x_1, \ldots, x_n$, it is clear that

$$\lim_{r\to\infty} \|\mathbf{K}_n - \mathbf{K}_n^{[r]}\| = 0. \tag{3.70}$$

Therefore, we can consider $\mathbf{K}_n$ to be a perturbation of $\mathbf{K}_n^{[r]}$. For this type of perturbation Weyl's Theorem bounds the perturbation in the eigenvalues. Combining these results, we obtain the following theorem:

Theorem 3.71 (Relative-Absolute Perturbation Bound) *Let k be a Mercer kernel function on $\mathcal{H}_\mu(\mathcal{X})$ with eigenvalues $(\lambda_i)_{i\in\mathbb{N}}$ and eigenfunctions $(\psi_i)_{i\in\mathbb{N}}$, and let $k^{[r]}$ be the truncated kernel function for some $r \in \mathbb{N}$. Let $\mathbf{K}_n$ and $\mathbf{K}_n^{[r]}$ be the induced kernel matrices for k and $k^{[r]}$, respectively, given n points $x_1, \ldots, x_n \in \mathcal{X}$. Then,*

$$|l_i - \lambda_i| \leq \begin{cases} \lambda_i \|\mathbf{\Psi}_n^{r\top}\mathbf{\Psi}_n^r - \mathbf{I}_r\| + \|\mathbf{K}_n - \mathbf{K}_n^{[r]}\|, & 1 \leq i \leq r \\ \lambda_i + \|\mathbf{K}_n - \mathbf{K}_n^{[r]}\| & r < i \leq n. \end{cases} \tag{3.72}$$

Consequently, since $\lambda_i \leq \lambda_r$ for $r < i \leq n$,

$$|l_i - \lambda_i| \leq \lambda_i \|\mathbf{\Psi}_n^{r\top}\mathbf{\Psi}_n^r - \mathbf{I}_r\| + \|\mathbf{K}_n - \mathbf{K}_n^{[r]}\| + \lambda_r. \tag{3.73}$$

Proof We want to apply Theorem 3.65 to the truncated kernel matrix. Introduce

$$\lambda_i^{[r]} = \begin{cases} \lambda_i, & 1 \leq i \leq r \\ 0, & r < i \leq n. \end{cases} \tag{3.74}$$

For $i \leq r$, by Theorem 3.65 since $\lambda_i = \lambda_i^{[r]}$,

$$|\lambda_i(\mathbf{K}_n^{[r]}) - \lambda_i| \leq \lambda_i \|\mathbf{\Psi}_n^{r\top}\mathbf{\Psi}_n^r - \mathbf{I}_r\|. \tag{3.75}$$

For $i > r$, since $\lambda_i(\mathbf{K}_n^{[r]}) = \lambda_i^{[r]} = 0$,

$$|\lambda_i(\mathbf{K}_n^{[r]}) - \lambda_i| = |\lambda_i| = \lambda_i. \tag{3.76}$$

Then,

$$\begin{aligned} |l_i - \lambda_i| &\leq |\lambda_i(\mathbf{K}_n^{[r]}) - \lambda_i| + |l_i - \lambda_i(\mathbf{K}_n^{[r]})| \\ &\leq \begin{cases} \lambda_i \|\mathbf{\Psi}_n^{r\top}\mathbf{\Psi}_n^r - \mathbf{I}_r\| + \|\mathbf{K}_n - \mathbf{K}_n^{[r]}\| & i \leq r, \\ \lambda_i + \|\mathbf{K}_n - \mathbf{K}_n^{[r]}\| & i > r, \end{cases} \end{aligned} \tag{3.77}$$

where the $|l_i - \lambda_i(\mathbf{K}_n^{[r]})|$ has been bounded by Theorem 3.49. ■

We see that the bound depends on the norms of two matrices. Set

$$\mathbf{C}_n^r = \mathbf{\Psi}_n^{r\top}\mathbf{\Psi}_n^r - \mathbf{I}_r, \qquad \mathbf{E}_n^r = \mathbf{K}_n - \mathbf{K}_n^{[r]}. \tag{3.78}$$

We will call $\|\mathbf{C}_n^r\|$ the *relative error term*, and $\|\mathbf{E}_n^r\|$ the *absolute error term*.

The transition from a degenerate kernel to a general kernel introduced the parameter r, which will be called *truncation point*. As already stated in Section 3.5, the truncation point r controls the number of leading eigenfunctions (these are the eigenfunctions of the leading eigenvalues sorted in non-increasing order) which enter the relative perturbation bound. Larger r will potentially result in a larger relative error, because more eigenfunctions are considered, and in a smaller absolute error term.

The relative-absolute bound from Theorem 3.71 holds for kernel matrices evaluated from any set of points $x_1, \ldots, x_n \in \mathcal{X}$. The next question is whether the error terms $\|\mathbf{C}_n^r\|$ and $\|\mathbf{E}_n^r\|$ converge to zero when the points $x_1, \ldots, x_n$ are given as an n-sample from μ, and $n \to \infty$.

First let us check that $\|\mathbf{C}_n^r\| \to 0$ almost surely for each r as $n \to \infty$. Recall that $\mathbf{\Psi}_n^r = (\psi_{i\ell}) \in \mathbb{M}_{n,r}$, with (see (3.67))

$$\psi_{i\ell} = \frac{1}{\sqrt{n}}\psi_\ell(X_i), \tag{3.79}$$

and $1 \le i \le n$, $1 \le \ell \le r$, and the (ψ_ℓ) constitute an orthonormal set of functions on the Hilbert space $\mathcal{H}_\mu(\mathcal{X})$, where μ is the common measure of the X_i. Then,

$$[\mathbf{\Psi}_n^{r\top}\mathbf{\Psi}_n^r]_{ij} = \frac{1}{n}\sum_{\ell=1}^{n}\psi_i(X_\ell)\psi_j(X_\ell). \tag{3.80}$$

Denoting by μ_n the empirical measure of n i.i.d. samples from μ, $\mu_n = \frac{1}{n}\sum_{i=1}^n \delta_{X_i}$, we can write the last equation as

$$[\mathbf{\Psi}_n^{r\top}\mathbf{\Psi}_n^r]_{ij} = \langle\psi_i, \psi_j\rangle_{\mu_n}. \tag{3.81}$$

By the strong law of large numbers, $\langle\psi_i, \psi_j\rangle_{\mu_n} \to \langle\psi_i, \psi_j\rangle_\mu = \delta_{ij}$ almost surely, and $\mathbf{\Psi}_n^{r\top}\mathbf{\Psi}_n^r \to \mathbf{I}_r$ element-wise. Since $\|\mathbf{\Psi}_n^{r\top}\mathbf{\Psi}_n^r - \mathbf{I}_r\| = \|\mathbf{C}_n^r\| \le r\max_{1\le i,j\le n}|c_{ij}|$, and the individual finitely many elements converge to zero, also $\|\mathbf{C}_n^r\| \to 0$ almost surely.

Now, in general, convergence of $\|\mathbf{C}_n^r\| \to 0$ can be arbitrarily slow, given no further restriction on the class of kernel functions. For example, fixing the norm of the eigenfunctions to 1, it is possible to increase the fourth moment $\mathsf{E}_\mu(\psi_i^2)$ beyond all limits such that the estimates of $\langle\psi_i, \psi_j\rangle_\mu$ have arbitrary large variance. Therefore, some form of regularity condition has to be imposed on the kernel function. We will discuss two cases. The first case, leading to tighter bounds, is given by Mercer kernels whose eigenfunctions are uniformly bounded, for example, when the eigenfunctions are a sine-basis (see Section 3.9). The second case, which is somewhat more relevant, is the case where the kernel function itself is bounded, but the individual eigenfunctions are not explicitly bounded. This case addresses for example the case of radial basis kernel functions. In the next two sections, we will study both cases in depth.

3.8 Relative-Absolute Bounds and Finite Precision Arithmetics

For non-degenerate kernel functions like the rbf-kernel (2.17), the integral operator T_k has infinitely many non-zero eigenvalues. The bound (3.72) tells us that we can expect approximation with relative precision only for a finite number of eigenvalues, whereas the remaining eigenvalues of the kernel matrix $\mathbf{K}$ obey absolute convergence bounds. While this might appear to be a drawback, it turns out that this picture is in fact accurate if one considers kernel matrices stored in computers with finite precision arithmetic like the ubiquitous floating-point formats.

In this setting, any real number can be stored only up to a certain precision. For 64-bit double precision floats as specified in (754-1985, 1985), the smallest number which can be added to 1 and still produce a different number is of the order of $\varepsilon = 10^{-16}$. This means that any kernel matrix

is stored in a slightly perturbed fashion, just as $\mathbf{K}$ is perturbed with respect to $\mathbf{K}^{[r]}$, leading to an additive perturbation of the spectrum of the order of ε. Thus, eigenvalues which are smaller than this level cannot be computed accurately. These observations explain the shape of the numerically computed eigenvalues as observed in Figures 3.1(c) and 3.1(d): The eigenvalues stagnate at a certain level due to the rounding errors.

Therefore, the eigenvalues of a matrix computed with finite precision arithmetic show the same properties as (3.72): below a certain level, the eigenvalues stagnate on a level which is typically of the order of the machine precision ε. Thus, the bound (3.72) gives an accurate picture of actual eigenvalues. This also gives a natural choice of r, namely r should be chosen such that $\lambda_r(\mathbf{K})$ is still larger than ε. Since $\|\mathbf{K} - \mathbf{K}^{[r]}\| \to 0$ as $r \to \infty$, such that for reasonable r, $\|\mathbf{K} - \mathbf{K}^{[r]}\|$ can become as small as ε.

3.9 Estimates I: Bounded Eigenfunctions

The first class of kernel functions we will consider are Mercer kernels whose eigenfunctions are uniformly bounded. Let

$$M_i = \operatorname*{ess\text{-}sup}_{x \in \mathcal{X}} |\psi_i(x)| = \|\psi_i\|_\infty, \qquad M = \sup_{i \in \mathbb{N}} M_i. \tag{3.82}$$

We require that $M < \infty$. An example for such a Mercer kernel is given as follows:

Example 3.83 Let $\mathcal{H}_\mathcal{X}(\mu)$ be the Hilbert space with $\mathcal{X} = [0, 2\pi]$, and μ is the uniform probability measure on $\mathcal{X}$ (meaning that for $0 \le a, b \le 2\pi$, $\mu([a, b]) = |a - b|/2\pi$. On this space, we consider the sine basis,

$$\varphi_i(x) = \sqrt{2}\sin(ix/2), \qquad i \in \mathbb{N}. \tag{3.84}$$

It holds that $\langle \varphi_i, \varphi_j \rangle_\mu = \delta_{ij}$. Obviously, $\|\varphi_i\|_\infty = \sqrt{2}$ independently of i.

Thus, if we consider $k(x, y)$ to be made up of the components $\lambda_i \varphi_i(x)\varphi_i(y)$, the scale of component i depends on the size of λ_i only. This situation will allow us to bound the approximation errors quite well. For such eigenfunctions we can derive a Hoeffding-type large deviation bound to estimate the size of $\|\mathbf{C}_n^r\|$:

Lemma 3.85 (Relative Error Term, Bounded Eigenfunctions) *Fix $r \in \mathbb{N}$. Let $\mathbf{C}_n^r = {\mathbf{\Psi}_n^r}^\top \mathbf{\Psi}_n^r - \mathbf{I}_r$ as in Theorem 3.71, and M be defined by* (3.82)*. Then, with probability larger than $1 - \delta$,*

$$\|\mathbf{C}_n^r\| < M^2 r \sqrt{\frac{2}{n} \log \frac{r(r+1)}{\delta}}. \tag{3.86}$$

Proof Denote the entries of $\mathbf{C}_n^r$ by $c_{\ell m} = [\mathbf{C}_n^r]_{\ell m}$. Then,

$$c_{\ell m} = \frac{1}{n} \sum_{i=1}^{n} \psi_\ell(X_i)\psi_m(X_i) - \delta_{\ell m}. \tag{3.87}$$

Note that

$$-M^2 - \delta_{\ell m} \le \psi_\ell(X_i)\psi_m(X_i) - \delta_{\ell m} \le M^2 - \delta_{\ell m}, \tag{3.88}$$

such that the range of $\psi_\ell(X_i)\psi_m(X_i)$ is given by $M^2 - \delta_{\ell m} + M^2 + \delta_{\ell m} = 2M^2$. Using the Hoeffding inequality (Theorem 2.40), it follows that

$$\mathsf{P}\left\{|c_{\ell m}| \ge \varepsilon\right\} \le 2\exp\left(-\frac{2n\varepsilon^2}{4M^4}\right). \tag{3.89}$$

In order to bound $\|\mathbf{C}_n^r\|$, recall that $\|\mathbf{C}_n^r\| \le r \max_{1 \le \ell, m \le r} |c_{\ell m}|$, thus

$$\mathsf{P}\left\{\|\mathbf{C}_n^r\| \ge \varepsilon\right\} \le \mathsf{P}\left\{\max_{1 \le \ell, m \le r} |c_{\ell m}| \ge \frac{\varepsilon}{r}\right\} \tag{3.90}$$

Since $c_{\ell m} = c_{m\ell}$, there are $r(r+1)/2$ different elements in the maximum, thus, by the union bound,

$$\mathbf{P}\left\{\max_{1\leq \ell,m\leq r} |c_{\ell m}| \geq \frac{\varepsilon}{r}\right\} \leq \sum_{\ell \geq m} \mathbf{P}\left\{|c_{\ell m}| \geq \frac{\varepsilon}{r}\right\} \leq r(r+1)\exp\left(-\frac{n\varepsilon^2}{2M^4 r^2}\right) \tag{3.91}$$

by Equation (3.89). Equating the right hand side with δ and solving (3.91) for ε results in the claimed inequality. ■

From this theorem, we see that for each fixed r, the convergence speed of $\|\mathbf{C}_n^r\| \to 0$ depends on the size r of $\mathbf{C}_n^r$ and M only. A relative bound for a larger number of eigenvalues will necessarily be less tight, but this effect is only due to the increased number eigenfunctions which are considered. In particular, the eigenvalues themselves do not appear in the bound.

Next we turn to the absolute error term which is governed by the truncation function e^r. Since the eigenfunctions are uniformly bounded,

$$|e^r(x,y)| = \left|\sum_{i=r+1}^{\infty} \lambda_i \psi_i(x)\psi_i(y)\right| \leq M^2 \sum_{i=r+1}^{\infty} \lambda_i \tag{3.92}$$

Therefore, if $\mathbf{E}_n^r = (e_{ij}) \in \mathbb{M}_n$, with $e_{ij} = e^r(X_i, X_j)/n$,

$$\|\mathbf{E}_n^r\| \leq n \max_{1\leq i,j\leq n} |e_{ij}| \leq M^2 \sum_{i=r+1}^{\infty} \lambda_i. \tag{3.93}$$

Using (3.93) and Theorem 3.85, we obtain the following relative-absolute bound.

Theorem 3.94 (Relative-Absolute Bound, Bounded Eigenfunctions)
Let k be a Mercer kernel on $\mathcal{H}_\mu(\mathcal{X})$ with eigenvalues $(\lambda_i)_{i\in\mathbb{N}}$ and eigenfunctions $(\psi_i)_{i\in\mathbb{N}}$. Let $\sup_i \|\psi_i\|_\infty = M < \infty$. Let μ be a probability measure on $\mathcal{X}$, and $\mathbf{K}_n$ be the normalized kernel matrix based on an n-sample from μ. Then, for $1 \leq r \leq n$, and $0 < \delta < 1$, with probability larger than $1-\delta$,

$$|\lambda_i(\mathbf{K}_n) - \lambda_i| \leq \lambda_i C(r,n) + E(r,n) \tag{3.95}$$

with

$$\begin{aligned} C(r,n) &< M^2 r \sqrt{\frac{2}{n}\log\frac{r(r+1)}{\delta}} \\ E(r,n) &< \lambda_r + M^2 \sum_{i=r+1}^{\infty} \lambda_i. \end{aligned} \tag{3.96}$$

In words, the eigenvalues of $\mathbf{K}_n$ converge to their limits with a relative-absolute bound, whose relative error term is independent of the eigenvalues, and increases almost linearly in r. The absolute error term is given by the sum of the remaining eigenvalues, which will be small if the eigenvalues decay quickly. We can thus say that the eigenvalues converge essentially on a relative scale with respect to the true eigenvalues of the kernel function.

For a more detailed asymptotic analysis, see Section 3.11.

3.10 Estimates II: Bounded Kernel Functions

The class of kernel functions with bounded eigenfunctions is rather restrictive, although it is possible to construct interesting learning algorithms using, for example, a kernel function based on the sine basis. An example which leads to unbounded eigenfunctions is the rbf-kernel (see (2.17)). Since the eigenfunctions can in principle become arbitrarily large (measured in the supremum norm or the fourth moment, while keeping the 2-norm fixed), we need to impose some form of

regularity condition. In this section, we assume that the "diagonal" $x \mapsto k(x,x)$ of the kernel function is bounded,

$$\sup_{x\in\mathcal{X}} |k(x,x)| = K < \infty. \tag{3.97}$$

This condition is quite natural for the class of kernels built from radial basis functions. These are kernel functions which can be written as

$$k(x,y) = g(\|x-y\|) \tag{3.98}$$

with an appropriate $g\colon \mathbb{R} \to \mathbb{R}$.

The first consequence of condition (3.97) is that the eigenfunctions and the error function e^r cannot become arbitrarily large.

Lemma 3.99 *Let k be a Mercer kernel with eigenvalues (λ_i) and eigenfunctions (ψ_i) such that the diagonal of k is uniformly bounded by K. Then for $I \subseteq \mathbb{N}$,*

$$0 \le \sum_{i\in I} \lambda_i \psi_i^2(x) \le k(x,x) \le K \tag{3.100}$$

for all $x \in \mathcal{X}$. In particular,

$$|\psi_i(x)| \le \sqrt{\frac{K}{\lambda_i}}. \tag{3.101}$$

Consequently, the error function e^r is bounded $0 \le e^r(x,x) \le K$ for all $r \in \mathbb{N}$.

Proof Since all the summands $\lambda_i\psi_i^2(x)$ are positive,

$$K \ge |k(x,x)| = \left|\sum_{i=1}^{\infty} \lambda_i\psi_i^2(x)\right| \ge \left|\sum_{i\in I} \lambda_i\psi_i^2(x)\right|. \tag{3.102}$$

The bound on ψ_i follows for $I = \{i\}$. The bound on e^r follows for $I = \{r+1,\dots\}$. ■

The error estimates will depend on certain regularity parameters of ψ_i and e^r. First of all, we are interested in the variance of $\psi_\ell\psi_m$ under μ since the expectation $\psi_\ell\psi_m$ is approximated via empirical means in $\mathbf{C}_n^r$. Using the standard notation that $\mathsf{E}_\mu(f)$ is the expectation of f with respect to the measure μ, and $\mathsf{Var}_\mu(f)$ the respective variance, define

$$\gamma_{\ell m}^2 = \mathsf{Var}_\mu(\psi_\ell\psi_m). \tag{3.103}$$

Moreover, we introduce the following expectation which is closely related to the absolute error term. For $r \in \mathbb{N}$, let

$$t_r = \mathsf{E}(e^r(X_1,X_1)). \tag{3.104}$$

3.10.1 The Relative Error Term

We begin by treating the relative error. The first step is to upper bound the variance of the random variables of which $\mathbf{C}_n^r$ is constructed.

Lemma 3.105 *Let (ψ_i) be the eigenfunctions of a Mercer kernel whose diagonal is uniformly bounded by K. Then, $\mathsf{E}_\mu(\psi_\ell^2\psi_m^2) \le \min(K/\lambda_\ell, K/\lambda_m)$, and*

$$\gamma_{\ell m}^2 = \mathsf{Var}_\mu(\psi_\ell\psi_m - \delta_{\ell m}) \le \min(K/\lambda_\ell, K/\lambda_m) - \delta_{\ell m}. \tag{3.106}$$

Proof By the Hölder inequality,

$$\mathsf{E}_\mu(\psi_\ell^2\psi_m^2) \le \|\psi_\ell^2\|_1 \|\psi_\ell^2\|_\infty \le \frac{K}{\lambda_\ell}, \tag{3.107}$$

because $\|\psi_\ell^2\|_1 = \|\psi_\ell\|_2^2 = 1$. The same bound holds with ℓ and m interchanged which proves the first inequality.

The second inequality follows from the definition of the variance and the fact that $\mathsf{E}_\mu(\psi_i\psi_j) = \delta_{ij}$:

$$\begin{aligned}\mathsf{Var}_\mu(\psi_\ell\psi_m - \delta_{\ell m}) = \mathsf{Var}_\mu(\psi_\ell\psi_m) &= \mathsf{E}_\mu(\psi_\ell^2\psi_m^2) - (\mathsf{E}_\mu\psi_\ell\psi_m)^2 \\ &\leq \min(K/\lambda_\ell, K/\lambda_m) - \delta_{\ell m},\end{aligned} \tag{3.108}$$

and the proof is completed. ■

Theorem 3.109 *Let k be a Mercer kernel with eigenvalues (λ_i) and let the diagonal of k uniformly bounded by K. Then, with probability larger than $1-\delta$,*

$$\|\mathbf{C}_n^r\| < r\sqrt{\frac{2K}{n\lambda_r}\log\frac{r(r+1)}{\delta}} + \frac{4rK}{3n\lambda_r}\log\frac{r(r+1)}{\delta} \tag{3.110}$$

Proof Let $c_{\ell m}$ be the entries of $\mathbf{C}_n^r$. It holds that

$$c_{\ell m} = \frac{1}{n}\sum_{i=1}^n \psi_\ell(X_i)\psi_m(X_i) - \delta_{\ell m}. \tag{3.111}$$

Therefore, for $1 \leq i \leq r$, by Lemma 3.99, $\sup_{x,y\in\mathcal{X}} |\psi_i(x)\psi_i(y)| \leq K/\lambda_r$,

$$-\frac{K}{\lambda_r} - \delta_{\ell m} \leq c_{\ell m} \leq \frac{K}{\lambda_r} - \delta_{\ell m}, \tag{3.112}$$

and the range of $c_{\ell m}$ has size $M := 2K/\lambda_r$.

We can bound the variance of $\psi_\ell(X_i)\psi_m(X_i) - \delta_{\ell m}$ using Lemma 3.105:

$$\mathsf{Var}_\mu(\psi_\ell\psi_m - \delta_{\ell m}) \leq \frac{K}{\lambda_r} =: \sigma^2. \tag{3.113}$$

By the Bernstein inequality (Theorem 2.42),

$$\mathsf{P}\left\{|c_{\ell m}| \geq \varepsilon\right\} \leq 2\exp\left(-\frac{n\varepsilon^2}{2\sigma^2 + 2M\varepsilon/3}\right) \tag{3.114}$$

In the proof of Theorem 3.85, we showed that

$$\mathsf{P}\left\{\|\mathbf{C}_n^r\| \geq \varepsilon\right\} \leq \sum_{\ell\geq m} \mathsf{P}\left\{|c_{\ell m}| \geq \frac{\varepsilon}{r}\right\}. \tag{3.115}$$

Thus,

$$\mathsf{P}\left\{\|\mathbf{C}_n^r\| \geq \varepsilon\right\} \leq r(r+1)\exp\left(-\frac{n(\varepsilon/r)^2}{2\sigma^2 + 2M\varepsilon/3r}\right). \tag{3.116}$$

Setting the right hand side equal to δ and solving for ε yields (compare Theorem 2.44) that with probability larger than $1-\delta$,

$$\|\mathbf{C}_n^r\| < \frac{2Mr}{3n}\log\frac{r(r+1)}{\delta} + r\sqrt{\frac{2\sigma^2}{n}\log\frac{r(r+1)}{\delta}}. \tag{3.117}$$

Substituting the values for σ^2 and M yields the claimed upper bound. ■

Remark 3.118 The previous lemma contains a term which scales as $O(1/n)$. However, using the Chebychev inequality, one can show that this term is not essential, because with probability larger than $1-\delta$,

$$\|\mathbf{C}_n^r\| < r\sqrt{\frac{r(r+1)K}{2\lambda_r n\delta}}. \tag{3.119}$$

Proof By the Chebychev inequality,

$$\mathbf{P}\left\{|c_{\ell m}| \geq \varepsilon\right\} \leq \frac{\mathbf{Var}_\mu(\psi_\ell\psi_m - \delta_{\ell m})}{n\varepsilon^2} \leq \frac{K}{\lambda_r n\varepsilon^2}. \tag{3.120}$$

Thus,

$$\mathbf{P}\left\{\|\mathbf{C}_n^r\| \geq \varepsilon\right\} \leq \frac{r(r+1)}{2}\frac{Kr^2}{\lambda_r n\varepsilon^2}. \tag{3.121}$$

Equating the right hand side to δ and solving for ε proves the remark. ■

Note however, that the scaling with δ and r is much worse for the bound based on the Chebychev inequality than for the bound based on the Bernstein inequality.

3.10.2 Absolute Error Term

Let us now turn to the absolute error term. The absolute error term in the relative-absolute bound (3.72) measures the size of $\mathbf{E}_n^r$ in the operator norm. Fortunately, the norm can be bounded rather efficiently since $\mathbf{E}_n^r$ is also positive definite because, by construction, e^r is also a Mercer kernel. Then, since the eigenvalues are all positive,

$$\|\mathbf{E}_n^r\| \leq \operatorname{trace}\mathbf{E}_n^r = \frac{1}{n}\sum_{i=1}^n e^r(X_i, X_i). \tag{3.122}$$

By the strong law of large numbers it follows that

$$\frac{1}{n}\sum_{i=1}^n e^r(X_i, X_i) \to_{\text{a.s.}} \mathbf{E}(e^r(X_1, X_1)) = t_r, \tag{3.123}$$

where t_r has been defined in (3.104). In the following, we will first relate t_r to the eigenvalues of k, compute certain statistical properties of e^r and then derive a finite sample size bound on $\|\mathbf{E}_n^r\|$.

We introduce the following handy notation for the tail sum of the eigenvalues:

$$\Lambda_{>r} = \sum_{i=r+1}^\infty \lambda_i. \tag{3.124}$$

First we show that t_r is actually equal to the tail sum of the eigenvalues.

Lemma 3.125 *Let k be a Mercer kernel with diagonal bounded by $K < \infty$ and eigenvalues (λ_i). Then,*

$$t_r = \sum_{i=r+1}^\infty \lambda_i = \Lambda_{>r}. \tag{3.126}$$

Proof Using (3.19), we compute

$$\begin{aligned} t_r &= \int_{\mathcal{X}} e^r(x,x)\mu(dx) = \int_{\mathcal{X}} \left(\sum_{i=r+1}^\infty \lambda_i\psi_i^2(x)\right)\mu(dx) \\ &= \sum_{i=r+1}^\infty \lambda_i \int_{\mathcal{X}} \psi_i^2(x)\mu(dx) = \sum_{i=r+1}^\infty \lambda_i\|\psi_i\|^2 = \sum_{i=r+1}^\infty \lambda_i = \Lambda_{>r}. \end{aligned} \tag{3.127}$$

Note that summation and integration commute because $\sum_{i=r+1}^R \lambda_i\psi_i^2(x)$ is bounded by K for all $R > r$, and Lebesgue's theorem. ■

Next, we compute statistical properties of $e^r(X_1, X_1)$ which are necessary for the application of the Bernstein inequality.

Lemma 3.128 *Let $\mathbf{E}_n^r$ be the truncation error matrix as defined in (3.78), and let $X \sim \mu$, the common distribution of the X_i. Then, for a kernel function with a diagonal uniformly bounded by K,*

$$0 \leq e^r(X, X) \leq K, \qquad \mathbf{Var}(e^r(X, X)) \leq K\mathbf{E}(e^r(X, X)) = Kt_r. \tag{3.129}$$

Proof The first inequality has already been proven in Lemma 3.99. With respect to the variance, note that

$$\begin{aligned} \mathbf{Var}(e^r(X, X) &= \mathbf{E}(e^r(X, X)^2) - (\mathbf{E}(e^r(X, X)))^2 \\ &\leq \mathbf{E}(|e^r(X, X)|) \sup_{x \in \mathcal{X}} |e^r(X, X)| - (\mathbf{E}(e^r(X, X)))^2 \\ &\leq \mathbf{E}(e^r(X, X)) \sup_{x \in \mathcal{X}} |e^r(X, X)| = t_r K \end{aligned} \tag{3.130}$$

by the Hölder inequality and Lemma 3.99. ■

We are now prepared to prove the error bound on $\mathcal{X}$.

Theorem 3.131 *Let k be a Mercer kernel on $\mathcal{H}_\mu(\mathcal{X})$ with eigenvalues (λ_i) and eigenfunctions (ψ_i), whose diagonal is bounded by $K < \infty$. Then, for a given confidence $0 < \delta < 1$, and $1 \leq r \leq n$ with probability larger than $1 - \delta$,*

$$\|\mathbf{E}_n^r\| < t_r + \sqrt{\frac{2Kt_r}{n} \log \frac{1}{\delta}} + \frac{2K}{3n} \log \frac{1}{\delta}. \tag{3.132}$$

Proof In Lemma 3.128, we have proven that the range of $e^r(X_i, X_i)$ has size K, and that $\mathbf{Var}(e^r(X_i, X_i)) \leq Kt_r$.

Thus, by the (one-sided) Bernstein inequality, with probability larger than $1 - \delta$,

$$\mathbf{P}\left\{\|\mathbf{E}_n^r\| - t_r \geq \varepsilon\right\} \leq \exp\left(-\frac{n\varepsilon^2}{2Kt_r + \frac{2K\varepsilon}{3}}\right).$$

Setting the right hand side equal to δ and solving for ε results in the claimed upper bound (compare (2.47)). ■

Remark 3.133 As in the case of the relative error term, using the Chebychev inequality, one can show that with probability larger than $1 - \delta$,

$$\|\mathbf{E}_n^r\| < t_r + \sqrt{\frac{Kt_r}{n\delta}}. \tag{3.134}$$

3.10.3 Relative-Absolute Bound for Bounded Kernel Functions

We can now derive the main result for bounded kernel functions. Combining Theorem 3.109, and Theorem 3.131, we obtain a final bound for the case where the kernel function is bounded.

Theorem 3.135 (Relative-Absolute Bound, Bounded Kernel Functions)
Let k be a Mercer kernel on $\mathcal{H}_\mu(\mathcal{X})$ with eigenvalues (λ_i), and a diagonal which is uniformly bounded by $K < \infty$. Let $\mathbf{K}_n$ be the normalized kernel matrix based on an n-sample from μ. Then, for $1 \leq r \leq n$, and $0 < \delta < 1$, with probability larger than $1 - 2\delta$,

$$|l_i - \lambda_i| \leq \lambda_i C(r, n) + E(r, n) \tag{3.136}$$

with

$$\begin{aligned} C(r,n) &< r\sqrt{\frac{2K}{n\lambda_r}\log\frac{2r(r+1)}{\delta}} + \frac{4Kr}{3n\lambda_r}\log\frac{2r(r+1)}{\delta}, \\ E(r,n) &< \lambda_r + \Lambda_{>r} + \sqrt{\frac{2K\Lambda_{>r}}{n}\log\frac{2}{\delta}} + \frac{2K}{3n}\log\frac{2}{\delta}. \end{aligned} \tag{3.137}$$

Proof The basic relative-absolute bounds holds by Theorem 3.71. The upper bounds on the relative error term $\|\mathbf{C}_n^r\|$ and the absolute error term $\|\mathbf{E}_n^r\|$ were derived in Theorem 3.109 and 3.131. The bound on $\|\mathbf{E}_n^r\|$ follows by Theorem 3.131 and substituting for t_r the term from Lemma 3.125.

Both bounds hold with probability larger than $1-\delta$. Therefore, combining both bounds with confidence $\delta/2$ using Lemma 2.57 leads to a bound on the sum which holds with probability $1-\delta$. ■

Remark 3.138 Again, note that the $O(1/n)$ terms in $C(r,n)$ and $E(r,n)$ are not essential. As stated in Remarks 3.118 and 3.133, using the Chebychev inequality, one can show that a similar bound holds with

$$C(r,n) = r\sqrt{\frac{r(r+1)K}{\lambda_r n\delta}}, \qquad E(r,n) = \lambda_r + \Lambda_{>r} + \sqrt{\frac{2K\Lambda_{>r}}{n\delta}}. \tag{3.139}$$

3.11 Asymptotic Considerations

In this section, we study the asymptotic rates of the error bounds for varying r and n. We will carry out these studies assuming two kinds of decay rates of the eigenvalues: polynomial and exponential.

Note that there are essentially two different strategies for using these bounds, depending on how r is chosen. As discussed in Section 3.8, a purely relative bound is unrealistic for eigenvalues computed using finite precision arithmetics. Therefore, it does not make sense to force $E(r,n)$ smaller than the precision ε of the arithmetics, such that r need not grow as $n \to \infty$. This means that one will usually keep r fixed, obtaining a relative bound for eigenvalues larger than ε and an absolute bound for the remaining eigenvalues on the order of ε. In this setting, only the asymptotic convergence speed of $C(r,n)$ with respect to n is interesting.

The other possibility is to let $r \to \infty$ as the number of samples tend to infinity, in order to obtain an asymptotically vanishing bound for all eigenvalues.

In this section, we will address both questions, by studying the asymptotic rates for $C(r,n)$ and $E(r,n)$, and determining a nearly optimal rate for $r(n)$ such that the overall bound tends to zero. In the following, we will always consider the asymptotic rates of the bounds we have obtained for fixed confidence δ, however, without always explicitly attaching the quantifier "with probability larger than $1-\delta$".

As a preparing step, we compute the asymptotic rates of certain tail sums.

Theorem 3.140 *For $\alpha > 1$,*

$$\sum_{i=r+1}^{\infty} i^{-\alpha} \leq \frac{r^{1-\alpha}}{\alpha-1} = O(r^{1-\alpha}). \tag{3.141}$$

For $\beta > 0$,

$$\sum_{i=r+1}^{\infty} e^{-\beta i} = \frac{e^{-\beta(r+1)}}{1-e^{-\beta}} = O(e^{-\beta r}). \tag{3.142}$$

Proof It holds that

$$\sum_{i=r+1}^{\infty} i^{-\alpha} \leq \int_{r+1}^{\infty} (x-1)^{-\alpha} dx = \int_{r}^{\infty} x^{-\alpha} dx = \left. \frac{x^{1-\alpha}}{1-\alpha} \right|_{r}^{\infty} = 0 - \frac{r^{1-\alpha}}{1-\alpha} = \frac{r^{1-\alpha}}{\alpha - 1}. \tag{3.143}$$

Since $\sum_{i=r+1}^{\infty} (e^{-\beta})^i$ is the tail of a geometric series,

$$\sum_{i=r+1}^{\infty} e^{-\beta i} \leq \frac{1}{1-e^{-\beta}} - \frac{1-\left(e^{-\beta}\right)^{r+1}}{1-e^{-\beta}} = \frac{\left(e^{-\beta}\right)^{r+1}}{1-e^{-\beta}}. \tag{3.144}$$

■

3.11.1 Bounded Eigenfunctions

In the case of bounded eigenfunctions, $C(r,n)$ does not depend on the eigenvalues, unlike $E(r,n)$, and its asymptotic rate with respect to r and n can be readily derived.

Corollary 3.145 *For bounded eigenfunctions, the asymptotic rate of $C(r,n)$ with respect to r and n is*

$$C(r,n) = O\left(n^{-\frac{1}{2}} r \sqrt{\log r}\right). \tag{3.146}$$

Proof For bounded eigenfunctions, we proved in Theorem 3.94 that

$$|l_i - \lambda_i| \leq \lambda_i C(r,n) + E(r,n) \tag{3.147}$$

with

$$C(r,n) < M^2 r \sqrt{\frac{2}{n} \log \frac{r(r+1)}{\delta}}, \tag{3.148}$$

with probability larger than $1-\delta$. The asymptotic rate of

$$C(r,n) = O\left(n^{-\frac{1}{2}} r \sqrt{\log r}\right) \tag{3.149}$$

can readily be derived. ■

The asymptotic rate of the absolute error term $E(r,n)$ depends on the decay rate of the eigenvalues. We consider two cases: polynomial and exponential decay.

Polynomial Decay

We assume that $\lambda_i = O(i^{-\alpha})$ for $\alpha > 1$.

Corollary 3.150 *For bounded eigenfunctions and polynomial decay of the eigenvalues, the asymptotic rate of $E(r,n)$ is given by*

$$E(r,n) = O\left(r^{1-\alpha}\right). \tag{3.151}$$

Proof By Theorem 3.94, it holds that $E(r,n) < \lambda_r + M^2 \sum_{i=r+1}^{\infty} \lambda_i$ with probability larger than $1-\delta$. The rate for $E(r,n)$ follows since by Theorem 3.140, $\sum_{i=r+1}^{\infty} \lambda_i = O(r^{1-\alpha})$. ■

Corollary 3.152 *For bounded eigenfunctions and polynomial decay of the eigenvalues, if $r(n) = cn^{1/2\alpha}$ for some $c > 0$, then the overall bound converges to zero with rate*

$$|l_i - \lambda_i| = O\left(n^{\frac{1-\alpha}{2\alpha}} \sqrt{\log n}\right). \tag{3.153}$$

Proof We wish to let r grow with n such that the bound from Theorem 3.94 tends to 0. By Corollaries 3.145 and 3.150, the rate for the approximation error $|l_i - \lambda_i|$ is

$$|l_i - \lambda_i| = O(\lambda_i r\sqrt{\log r} n^{-\frac{1}{2}} + \Lambda_{>r}). \tag{3.154}$$

For our considerations, the $\sqrt{\log r}$ term can be neglected. From $\lambda_i = O(i^{-\alpha})$, we obtain the following condition:

$$rn^{-\frac{1}{2}} + r^{1-\alpha} = o(1). \tag{3.155}$$

We use the following Ansatz: let $r = n^{\varepsilon}$ with $\varepsilon > 0$. Thus, we wish to find ε such that

$$n^{\varepsilon-\frac{1}{2}} + n^{\varepsilon(1-\alpha)} = o(1). \tag{3.156}$$

This condition is obviously met if $\varepsilon < 1/2$. We wish to balance the two terms in order to minimize the overall rate. This rate is attained if

$$\varepsilon - \frac{1}{2} = \varepsilon(1-\alpha) \quad \rightsquigarrow \quad \varepsilon = \frac{1}{2\alpha}. \tag{3.157}$$

Plugging in this rate shows that

$$|l_i - \lambda_i| = O(n^{\frac{1-\alpha}{2\alpha}}\sqrt{\log n}) \tag{3.158}$$

and the proof of the corollary is completed. ■

Exponential Decay

Here, we assume that $\lambda_i = O(e^{-\beta i})$ for $\beta > 0$.

Corollary 3.159 *For bounded eigenfunctions and exponential decay of the eigenvalues, it holds that*

$$E(r,n) = O\left(e^{-\beta r}\right). \tag{3.160}$$

Proof Again by Theorem 3.140, the rate of $E(r,n)$ can be readily computed. ■

Corollary 3.161 *For bounded eigenfunctions and exponential decay of the eigenvalues, choosing* $r(n) = \log(cn^{1/2\beta})$ *for some* $c > 0$ *leads to an overall asymptotic rate of*

$$|l_i - \lambda_i| = O\left(n^{-\frac{1}{2}}(\log n)^{\frac{3}{2}}\right). \tag{3.162}$$

Proof We want to determine the slowest rate for $r(n) \to \infty$ such that the rate of $\Lambda_{>r}$ is not slower than $O(n^{-\frac{1}{2}})$. Using the Ansatz $r = \log n^{\varepsilon}$, we obtain the condition

$$e^{-\beta \log n^{\varepsilon}} = n^{-\beta\varepsilon} = O(n^{-\frac{1}{2}}) \qquad \text{if} \qquad -\beta\varepsilon \leq -\frac{1}{2} \quad \rightsquigarrow \quad \varepsilon = \frac{1}{2\beta}. \tag{3.163}$$

Plugging this choice of ε gives the overall rate of

$$|l_i - \lambda_i| = O(n^{-\frac{1}{2}}(\log n)^{\frac{3}{2}}) \tag{3.164}$$

which concludes the proof of the corollary.

■

3.11.2 Bounded Kernel Functions

We turn to the case of bounded kernel functions. We will discuss only the more realistic and simpler bound based on the Chebychev inequality, because it is this bound one would use in practical situations.

We are again interested in the asymptotic behavior with respect to r and n.

Corollary 3.165 *For bounded kernel functions, the asymptotic rates of $C(r,n)$ and $E(r,n)$ with respect to r and n for eigenvalues (λ_i) are given by*

$$C(r,n) = O\left(\lambda_r^{-\frac{1}{2}} r^2 n^{-\frac{1}{2}}\right), \qquad E(r,n) = O\left(\Lambda_{>r} + \sqrt{\Lambda_{>r}} n^{-\frac{1}{2}}\right). \tag{3.166}$$

Proof For the terms $C(r,n)$ and $E(r,n)$, we have derived the following bounds (see (3.139))

$$C(r,n) < r\sqrt{\frac{r(r+1)K}{\lambda_r n\delta}}, \qquad E(r,n) < \lambda_r + \Lambda_{>r} + \sqrt{\frac{2K\Lambda_{>r}}{n\delta}}. \tag{3.167}$$

From these bounds, one can readily deduce the claimed asymptotic rates. ■

As in the previous section, we consider the two cases of polynomial and exponential decay of the true eigenvalues.

Polynomial Decay

We assume that $\lambda_r = O(r^{-\alpha})$ for $\alpha > 1$.

Corollary 3.168 *For bounded kernel functions and polynomial decaying eigenvalues, we obtain the rates*

$$C(r,n) = O\left(r^{2+\frac{\alpha}{2}} n^{-\frac{1}{2}}\right), \qquad E(r,n) = O\left(r^{1-\alpha} + r^{\frac{1-\alpha}{2}} n^{-\frac{1}{2}}\right). \tag{3.169}$$

For the choice

$$r(n) = cn^{\frac{1}{2+3\alpha}} \tag{3.170}$$

for some $c > 0$, one obtains the asymptotic rate

$$|l_i - \lambda_i| = O\left(n^{\frac{1-\alpha}{2+3\alpha}}\right). \tag{3.171}$$

Proof The two rates for $C(r,n)$ and $E(r,n)$ again follow by plugging in the estimates for the tail sums of the eigenvalues from Theorem 3.140.

With respect to the rate for $r(n) \to \infty$, plugging in $\lambda_r = r^{-\alpha}$, $\Lambda_{>r} = r^{1-\alpha}$ (omitting the constants) gives

$$r^{2+\frac{\alpha}{2}} n^{-\frac{1}{2}} + r^{1-\alpha} + r^{\frac{1-\alpha}{2}} n^{-\frac{1}{2}}. \tag{3.172}$$

We set $r = n^{\varepsilon}$ and obtain the sum

$$n^{\varepsilon(2+\frac{\alpha}{2})-\frac{1}{2}} + n^{\varepsilon(1-\alpha)} + n^{\varepsilon(\frac{1-\alpha}{2})-\frac{1}{2}}. \tag{3.173}$$

Of these, the first and second term are essential. They are balanced if

$$\varepsilon\left(2+\frac{\alpha}{2}\right) - \frac{1}{2} = \varepsilon(1-\alpha) \quad \rightsquigarrow \quad \varepsilon = \frac{1}{2+3\alpha}. \tag{3.174}$$

Plugging this into either term yields the claimed rate for the approximation error. ■

Exponential Decay

We assume that $\lambda_r = O(e^{-\beta r})$ for $\beta > 0$.

Corollary 3.175 *For bounded kernel functions and exponentially decaying eigenvalues, it holds that*

$$C(r,n) = O\left(e^{\frac{\beta}{2}r} r^2 n^{-\frac{1}{2}}\right), \qquad E(r,n) = O\left(e^{-\beta r} + e^{-\frac{\beta}{2}r} n^{-\frac{1}{2}}\right). \tag{3.176}$$

Setting

$$r(n) = \log c n^{\frac{1}{3\beta}} \tag{3.177}$$

for some $c > 0$ gives the asymptotic rate

$$|l_i - \lambda_i| = O\left(n^{-\frac{1}{3}} (\log n)^2\right). \tag{3.178}$$

Proof The two rates follow by plugging in the estimates for the tail sums of the eigenvalues from Theorem 3.140

Now in order to obtain the asymptotic rate for $r(n)$, note that in this case, $\lambda_r = O(e^{-\beta r})$, $\Lambda_{>r} = O(e^{-\beta r})$. Therefore, the rate (omitting all constants) becomes

$$e^{\frac{\beta}{2}r} r^2 n^{-\frac{1}{2}} + e^{-\beta r} + e^{-\frac{\beta}{2}r} n^{-\frac{1}{2}}. \tag{3.179}$$

With the Ansatz $r = \log n^{\varepsilon}$, we get

$$n^{\frac{\beta\varepsilon}{2} - \frac{1}{2}} (\log n^{\varepsilon})^2 + n^{-\beta\varepsilon} + n^{-\frac{\beta\varepsilon}{2} - \frac{1}{2}}. \tag{3.180}$$

From the first term we get that $\varepsilon \leq 1/\beta$, otherwise it diverges. But for $\varepsilon \leq 1/\beta$, the third term is always smaller than the second term, such that we have to balance the first and the second term. Thus, the optimal rate is given if

$$\frac{\beta\varepsilon}{2} - \frac{1}{2} = -\beta\varepsilon \quad \rightsquigarrow \quad \varepsilon = \frac{1}{3\beta}. \tag{3.181}$$

This choice results in the claimed rate for the approximation error. ■

3.12 Examples

We consider some examples. These examples should provide some insight into the shape of the bounds and in how well the bounds approximate the true errors.

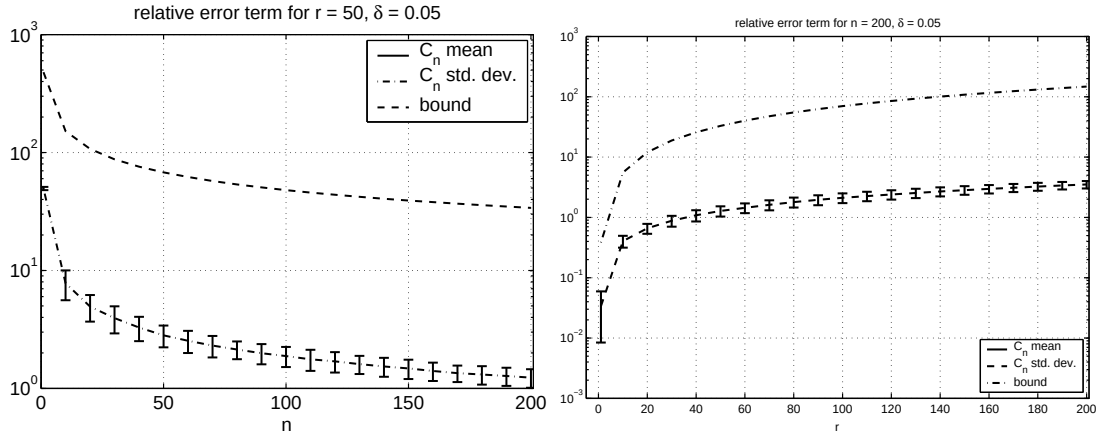
3.12.1 Examples: Bounded Eigenfunctions

We compare the theoretical bound with the actual error terms and approximation errors for the case of the sine-basis example. The sine-basis example was given as follows: The underlying Hilbert space is given by $[0, 2\pi]$ equipped with the uniform probability measure. The basis functions are given by (compare (3.84))

$$\varphi_i(x) = \sqrt{2}\sin(ix/2), \qquad i \in \mathbb{N}. \tag{3.182}$$

These functions are uniformly bounded by $M = \sqrt{2}$.

Figure 3.3 shows a plot of the bound and sampled norms of $\mathbf{C}_n^r$. In Figure 3.3(a), the truncation point r is kept fixed, while the number of samples is varied. We see that $\|\mathbf{C}_n^r\|$ basically decays as $1/\sqrt{n}$, both in the bound and the actual norm as predicted by Corollary 3.145, although the bound is roughly off by an order of magnitude. In Figure 3.3(b), the number of sample points is kept constant and the truncation point is varied. Here, we see the effect of computing the relative error term for a larger number of eigenvalues very clearly. Again, the bound shows roughly the same rate, although it is again off by an order of magnitude.



(a) Sampled $C(r, n)$ and the bound from (3.86) for varying n.

(b) The same quantities as in (a) for varying r. We see that the predicted increase of the error for larger r matches the observed increase.

Figure 3.3: Relative error term $C(r, n)$ for a kernel function with bounded eigenfunctions (sine basis based Mercer kernel).

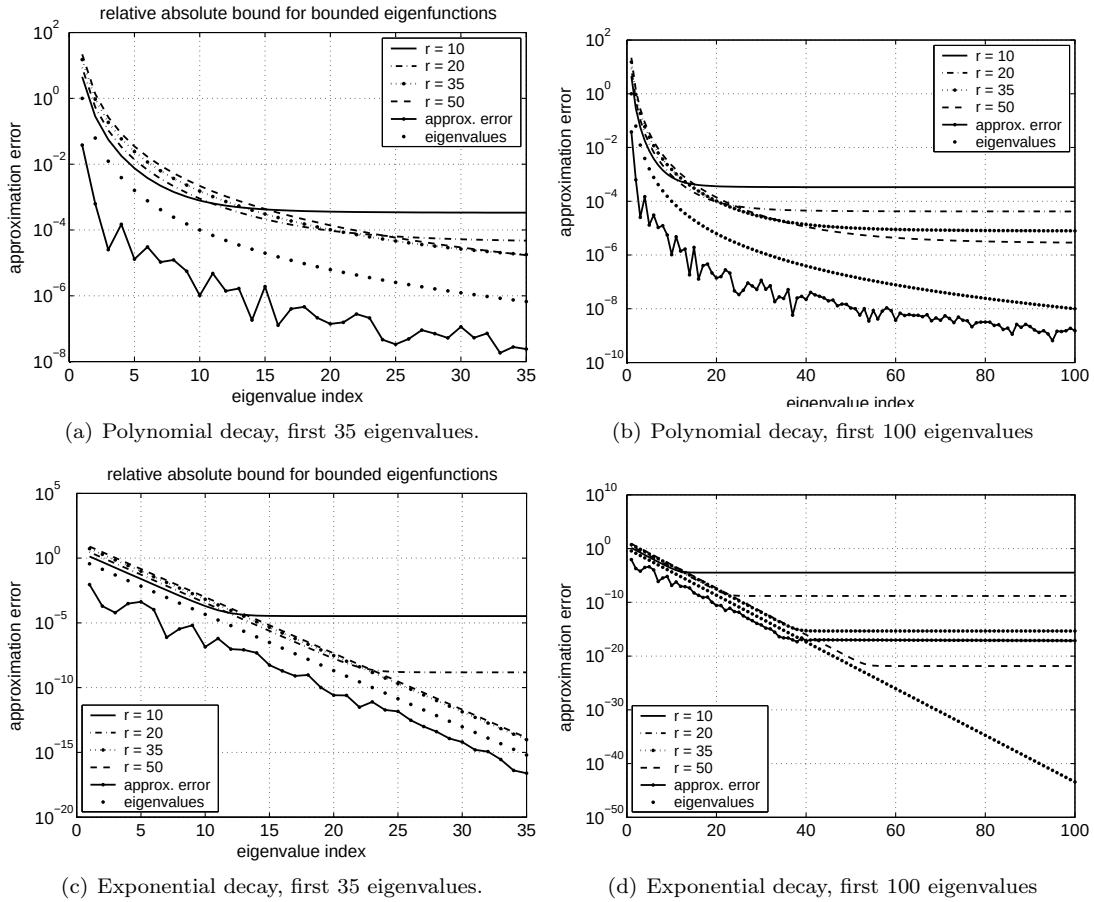
Next, we want to compare the bound to the actual approximation errors of the eigenvalues. We again discuss the two cases of polynomial and exponential decay of the eigenvalues. More specifically, we consider the cases $\lambda_i = i^{-4}$ and $\lambda_i = e^{-i}$.

Unfortunately, we cannot compute the resulting kernel function in closed form. Instead, we truncate the kernel function to the first 1000 terms, such that the difference will be negligible. We sample $n = 500$ points and compute the eigenvalues of the associated kernel matrix. The first 35 and 100 eigenvalues are plotted in Figure 3.4 together with the relative absolute bound (3.95). We see that there is no r such that the resulting bound is smaller than all the others. For larger r, the bounds become much smaller for large i, but at the same time, the bound becomes larger for small i. Therefore, one really has to consider the whole family of bounds to obtain a tight estimate. Apart from that, the lower hull of all these bounds reflects the size of the error quite well.

We can also clearly see the effect due to finite-precision arithmetics discussed in Section 3.8. For the case of exponential decay, the error stops decreasing at around eigenvalue λ_{40} and begins to stagnates around 10^{-18}. Moreover, this effect is not captured by the bound: for $r = 50$, the bound is already smaller than the experimentally measured error. This effect is due to the finite-precision arithmetic used to calculate these experiments. We see that the absolute error term is not merely an artifact of our derivations, but that in a real setting involving matrices stored with finite-precision arithmetics, a pure relative error bound is not possible due to perturbations coming from round-off errors. For $r = 35$, the bound actually matches the observed errors quite well.

In summary, we see that the bounds on the relative error term capture the convergence speed quite well while being off by one order of magnitude. This overestimation of the error is most likely due to the use of the union bound over all entries of the matrix. Recall that the union bound is tight only if the individual events are disjoint. This will not be the case for the entries of the relative error matrix. Unfortunately, no better estimate is easily available.

The resulting bounds are nevertheless quite tight, and in particular correctly predict that the error scales with the magnitude of the eigenvalue. This has to be contrasted with a purely absolute error bound which would be of the size of the largest approximation error.



(a) Polynomial decay, first 35 eigenvalues.

(b) Polynomial decay, first 100 eigenvalues

(c) Exponential decay, first 35 eigenvalues.

(d) Exponential decay, first 100 eigenvalues

Figure 3.4: Approximation error and the relative-absolute bound for different truncation errors for the example with bounded eigenfunctions (sine basis). The number of samples was $n = 500$. In the upper row, the true eigenvalues are $\lambda_i = i^{-4}$, while in the lower row, $\lambda_i = \exp(-i)$. One can clearly see that for varying r, there is a trade-off between the size of the absolute error term and the relative error term: For smaller r, the absolute error term becomes larger while the bound for the leading eigenvalues is smaller. One can also see that due to the finite precision arithmetics, the eigenvalues stagnate at around 10^{-18}. For the lower row, $r = 35$ results in an upper bound which accurately reflects the true structure of the eigenvalues.

3.12.2 Bounded Kernel Functions: An extremal example

We want to assess the quality of the bounds for bounded kernel functions from Section 3.10. Unlike in the case of bounded eigenfunctions, we have not yet developed an example. In this section we first develop an example which is extremal in the sense that the upper bound from Lemma 3.99 is actually achieved, which means that the eigenfunctions actually become as large as possible given that the kernel function itself is bounded.

We require that $k(x,x) = 1$ for all $x \in \mathcal{X}$. From the proof of Lemma 3.99, we see that ψ_i becomes maximal if for each x, there exists only one index i such that ψ_i is non-zero. Therefore, let $(A_i)_{i\in\mathbb{N}}$ be a partition of $\mathcal{X}$. Then, let

$$\lambda_i = \mu(A_i), \qquad \text{and} \qquad \psi_i(x) = \frac{1}{\sqrt{\lambda_i}} 1_{A_i}(x). \tag{3.183}$$

The associated Mercer kernel is

$$k(x,y) = \sum_{i=1}^{\infty} 1_{A_i}(x) 1_{A_i}(y) = \begin{cases} 1 & \text{if there exists an } i \text{ such that } x,y \in A_i, \\ 0 & \text{else.} \end{cases} \tag{3.184}$$

For the viewpoint of machine learning applications, note that functions $\hat{f}(x) = \sum_{i=1}^{n} k(x, X_i)\alpha_i$ with $\alpha \in \mathbb{R}^n$ are piecewise constant on each A_i. Therefore, using this kernel for kernel machine learning methods results in a rather inflexible hypothesis space.

Let us compute the error terms. Although we could just use the general results from Section 3.9, let us take advantage of the fact that the parameters can be computed in closed form. This is an example of how the bounds can be improved in the presence of additional information, and it also allows us to check how realistic the general estimates are. We begin with computing $\|\mathbf{C}_n^r\|$. For this quantity, we have to study $\psi_i(X_1)\psi_j(X_1)$. This function is

$$\psi_i(X_1)\psi_j(X_1) = \frac{1}{\sqrt{\lambda_i \lambda_j}} 1_{A_i}(X_1) 1_{A_j}(X_1). \tag{3.185}$$

First, since A_i and A_j are disjoint, $\psi_i(X_1)\psi_j(X_1) = 0$ if $i \neq j$. For $i = j$, ψ_i^2 is either $1/\lambda_i$ or 0, and

$$\mathbf{P}\left\{\psi_i^2(X_1) = \frac{1}{\lambda_i}\right\} = \mathbf{P}\left\{X_1 \in A_i\right\} = \lambda_i. \tag{3.186}$$

Thus, the expectation of ψ_i^2 is 1 by construction. Let us compute the variance of ψ_i^2. Since ψ_i^4 takes the values 0 and $1/\lambda_i^2$, the expectation is

$$\frac{1}{\lambda_i^2}\mathbf{P}\left\{\psi_i^4(X_1) = \frac{1}{\lambda_i^2}\right\} + 0 = \frac{1}{\lambda_i^2}\lambda_i = \frac{1}{\lambda_i}. \tag{3.187}$$

Thus,

$$\mathbf{Var}_{\mu}(\psi_i^2) = \frac{1}{\lambda_i} - 1 \leq \frac{1}{\lambda_i}. \tag{3.188}$$

Note that this achieves the upper bound from Lemma 3.105 for the case where $i = j$.

Combining these observations, we obtain that $\mathbf{C}_n^r = \operatorname{diag}(c_1, \ldots, c_n)$ with

$$c_i = \frac{1}{n}\sum_{\ell=1}^{n} \psi_i^2(X_\ell) - 1. \tag{3.189}$$

Since we will be looking at numerical simulations, the sample size will be relatively small. Therefore, we prefer the bound based on the Chebychev inequality. Moreover, although the distribution of c_i is completely known, we use the Chebychev inequality to be able to compare this result with the general bound. Thus,

$$\mathbf{P}\left\{|c_i| \geq \varepsilon\right\} \leq \frac{1}{\lambda_i n \varepsilon^2}. \tag{3.190}$$

Since $\mathbf{C}_n^r$ is diagonal, the eigenvalues of $\mathbf{C}_n^r$ are given by the diagonal elements. Thus, $\|\mathbf{C}_n^r\| = \max_{1\leq i\leq r} |c_i|$, and

$$\mathbf{P}\left\{\max_{1\leq i\leq r} |c_i| \geq \varepsilon\right\} \leq \frac{r}{\lambda_r n\varepsilon^2}, \tag{3.191}$$

by the union bound. Then, with probability larger than $1-\delta$,

$$\|\mathbf{C}_n^r\| = \max_{1\leq i\leq r} |c_i| < \sqrt{\frac{r}{\lambda_r n\delta}}. \tag{3.192}$$

If we plug this estimate into the relative approximation bound from Theorem 3.65, obtain

$$|\lambda_i(\mathbf{K}_n) - \lambda_i| \leq \lambda_i\|\mathbf{C}_n\| < \lambda_i\sqrt{\frac{r}{\lambda_r n\delta}} = \frac{\lambda_i}{\sqrt{\lambda_r}}\sqrt{\frac{r}{n\delta}}. \tag{3.193}$$

In other words, the penalty for computing the relative bound for a larger number of eigenvalues is actually rather severe if the eigenvalues decay quickly. We will discuss the consequences of this fact later.

In Equation (3.192) we obtained an estimate of $\mathbf{C}_n^r$ analogous to the one from Remark 3.118. However, a significant difference is that the estimate for the relative error term scales with $\sqrt{r}$ in r, whereas the general result scales as r^2. This difference is due to the fact that in this example, $\mathbf{C}_n^r$ is diagonal. In the general case, we cannot a priori assume that the off-diagonal elements are zero, resulting in a less tight bound. But in summary we see that there actually exist functions such that the relative error term contains the factor $1/\sqrt{\lambda_r}$, and that this factor occurred not merely due to technical artifacts of the derivation.

Figure 3.5 plots $\|\mathbf{C}_n^r\|$ for this kind of indicator function eigenfunctions. We sampled X_i uniformly from $[0,1]$, and chose the eigenvalues as

$$\lambda_i = \frac{1}{Z}\exp(-i/20), \tag{3.194}$$

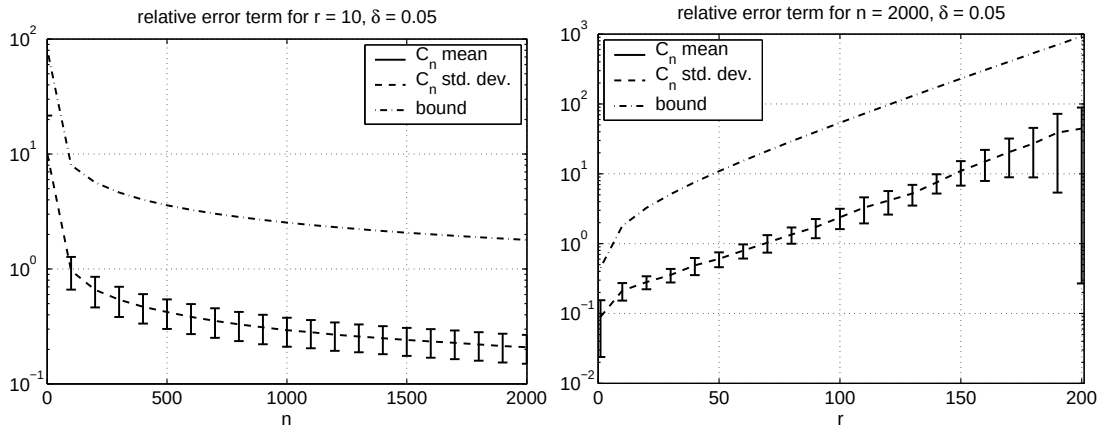
with $Z = \sum_{i=1}^{\infty}\exp(-i/20) = \frac{e^{-1/20}}{1-e^{-1/20}}$ (to meet the normalization condition), to ensure that the eigenvalues get fairly small. In Figure 3.5(a), $\|\mathbf{C}_n^r\|$ is plotted for fixed r and varying n to show the decay rate with increasing sample-size. Both the bound and the actual data show roughly the same rate, although the bound is off by a factor of 10. Again, this is due to the union bound, but since the events are not disjoint, and no further information is available, no better bound is available. In Figure 3.5(b), we see the effect of increasing the truncation point r. We showed that with probability larger than $1-\delta$,

$$\|\mathbf{C}_n^r\| \leq \sqrt{\frac{r}{\lambda_r n\delta}}. \tag{3.195}$$

In contrast to the bound for bounded eigenfunctions, this bound depends on the eigenvalues of the kernel function. Now from (3.195), we expect that $\|\mathbf{C}_n^r\| = O(\sqrt{r/\lambda_r})$ with varying r. This means that in a semi-logarithmic plot, we can expect to see a more or less straight line because

$$\log\sqrt{r/\lambda_r} = \frac{1}{2}\left(\log r + \log Z - \log(\exp(-r/20)\right) = O(r+\log r). \tag{3.196}$$

Figure 3.5(b) depicts the experimental results. We see that the empirically measured error actually increases in roughly the same rate as predicted by the theory. Although this might not hold in general, we have seen that the asymptotic rates predicted by the bound can actually be achieved for certain set of eigenvalues and eigenfunctions. Therefore, we can support the observation that in the case of bounded kernel functions (and unbounded eigenfunctions), the relative error depends heavily on the truncation point r and can in fact become very large given that the eigenvalues decay quickly enough.



(a) Sampled relative approximation error $C(r,n)$ and the bound (3.110) for varying n.

(b) The same quantities as in (a), but for varying r. As predicted by the bound, larger r lead to much larger errors.

Figure 3.5: Relative error term $C(r,n)$ for the bounded kernel function from Section 3.10.

This situation is likely even more severe in the general case where we have shown in Theorem 3.165:

$$\|\mathbf{C}_n^r\| = O(n^{-\frac{1}{2}} r^2 \lambda_r^{-\frac{1}{2}}). \tag{3.197}$$

The increase from $\sqrt{r}$ to r^2 is due to the fact that for general eigenfunctions, $\mathbf{C}_n$ need not be diagonal, so we have to consider a larger number of random variables, leading to a larger constant via the union bound.

Next we turn to the relative-absolute error bounds. This time, we consider the following eigenvalues:

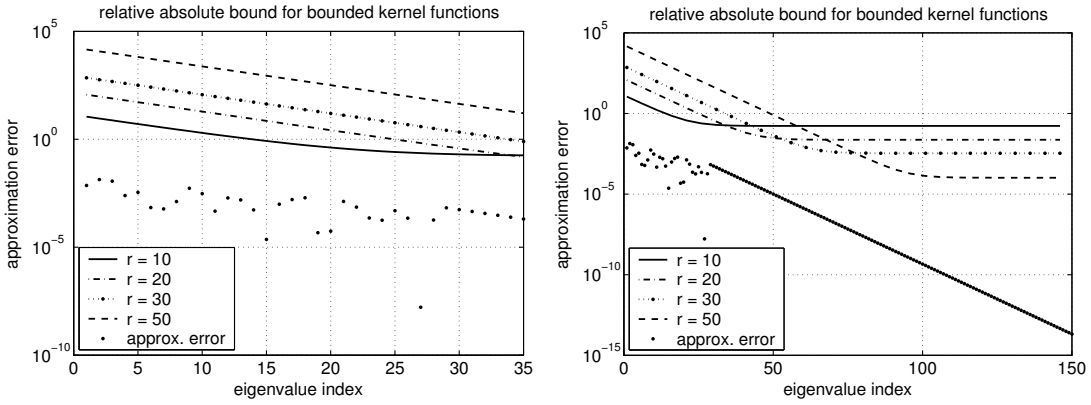
$$\lambda_i = \frac{1}{Z}\exp(-i/5). \tag{3.198}$$

Since the λ_i must sum to one, Z is the corresponding normalization constant:

$$Z = \sum_{i=1}^{\infty} \lambda_i = \frac{e^{-1/5}}{1-e^{-1/5}}. \tag{3.199}$$

We again compute the kernel truncated to the first 1000 terms based on a sample of size 1000 uniformly drawn from $[0,1]$. Actually, we have two different relative-absolute bounds, the first being the general relative-absolute bound from Theorem 3.135. The second bound is obtained by replacing the estimate for the relative error term by (3.192) which has been computed for the special eigenfunction set used in this example.

Figure 3.6 plots the general bound while Figure 3.7 plots the special bound using the tighter estimate. Not surprisingly, the specially adapted bound is much tighter, but we see that both bounds reflect the fact that the approximation error decreases according to the size of the eigenvalue. In fact, the measured approximation error decays roughly with the rate predicted by the bounds up to eigenvalue λ_{35}, after which the approximation error becomes a straight line decaying at twice the original speed. The reason for this effect is as follows. For small eigenvalues, the eigenfunctions take very large values but also only on a very small set. Therefore, for the number of samples chosen in the experiments ($n = 1000$), all eigenfunctions for extremely small eigenvalues are zero on all the X_i with high probability, such that the kernel matrix has effectively only finite rank (smaller than n), and the associated eigenvalues are zero. Then, the approximation error is $|\lambda_i(\mathbf{K}_n) - \lambda_i| = \lambda_i$. Therefore, the approximation error becomes a straight line which decays quicker than the $\sqrt{\lambda_i}C$ upper bound on the error. However, the rate $\sqrt{\lambda_i}C$ holds for the first few leading eigenvalues. In summary, this means that the factor $1/\sqrt{\lambda_r}$ is not an artifact of the



(a) first 35 eigenvalues (b) first 150 eigenvalues

Figure 3.6: Approximation error and the relative-absolute bound for different truncation errors for the example with bounded kernel function. In this plot, the general bound from Theorem 3.135 for bounded kernel functions is used. The sample size was $n = 1000$ and the eigenvalues were $\lambda_i = \exp(-i/5)/Z$, where Z is a normalization constant such that the eigenvalues sum to 1.

derivation, but that there actually exist kernels whose eigenvalues converge with the slower rate of $\sqrt{\lambda_r}$.

We also again see that the error stagnates at around 10^{-18}. This is again the effect of finite-precision arithmetics. Furthermore, we see that with increasing r, the bound becomes larger for small i. This effect is more prominent for the general bound, because the relative-error term scales much faster with r (r^2 as opposed to $\sqrt{r}$).

In summary, we can say that the relative-absolute bound reflects the fact quite well that the estimation error for smaller eigenvalues is much smaller than that of large eigenvalues. We have also seen that these tight bounds can only be obtained by using the whole family of bounds (using all bounds for $1 \leq r \leq n$). If for some reason one wants to use just one bound, we suggest either setting $r = i$, or fixing i at a level such that the absolute error term is small enough for whatever application one has in mind. Using $r = i$ shows that the error can be bounded roughly by $\lambda_i C + E$, for the case of bounded eigenfunctions, and $\sqrt{\lambda_i} C + E$, for bounded kernel functions, where E is governed by the sum of all eigenvalues which are smaller than λ_i.

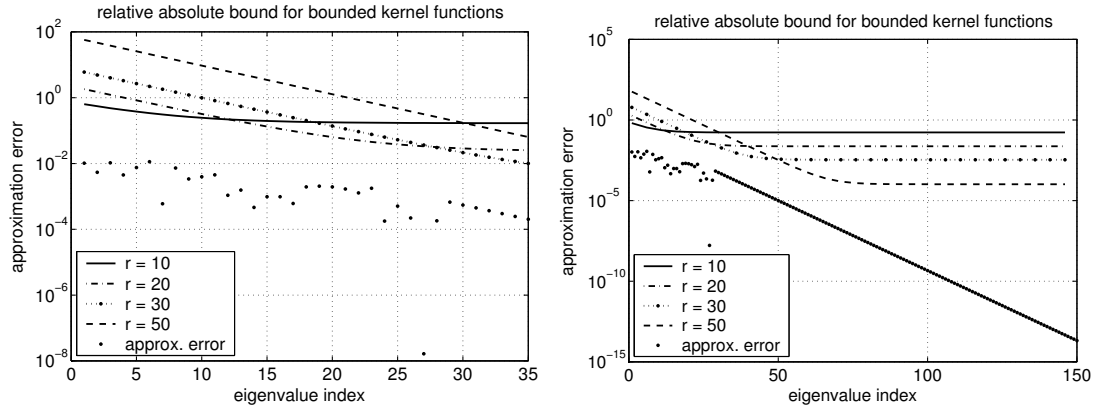
3.13 Discussion

We conclude this chapter with a discussion of the results. We split the discussion into three parts, the actual relative-absolute bound, and the two classes of kernel functions, those with bounded eigenfunctions and bounded kernel functions.

3.13.1 The Relative-Absolute Bound

The relative-absolute bound we have derived in Theorem 3.71 follows a considerably different approach than the line of research which is based on the variational characterization of the eigenvalues (see, for example (Shawe-Taylor et al., 2002a), (Zwald et al., 2004)). There, the approach directly considers concentration inequalities for a functional which is an alternative definition of the eigenvalues. Here, in contrast, the basic bound is first derived in a purely algebraic setting. Then, the size of several error matrices are estimated in a probabilistic setting.

The approach also differs considerably from the functional analytic approach taken for example by von Luxburg (2004), because the convergence is considered in a finite dimensional setting, and not by embedding the operator represented by the kernel matrix into a function space.



(a) first 35 eigenvalues (b) first 150 eigenvalues

Figure 3.7: The same plot as Figure 3.7, but here, the more special estimate (3.192) is used for the relative error term, resulting in a better estimate of the approximation error.

Compared to the existing approaches, the current approach is perhaps most similar to the one taken by Koltchinskii and Giné (2000). There, convergence is also considered in a finite-dimensional setting. However, the approach taken in these works uses the Hoffman-Wielandt inequality, which measures the error on an absolute scale. Moreover, that work aims at proving asymptotic results in the form of laws of large numbers and central limit theorems, whereas in this work, we aim at finite sample size confidence bounds.

As already stated before, a significant feature of the approximation error bounds developed in this chapter is that they scale with the magnitude of the true eigenvalue. All current finite sample size bounds treat the error on a fixed scale, such that the error estimate is governed by the largest error, which is usually given by the largest eigenvalues. Therefore, the new bounds provide a much more accurate picture of the approximation errors.

Finitely, note that the different approaches lead to different measures of the approximation error. The approach based on the variational characterization of the eigenvalues naturally leads to differences in sums of eigenvalues. The approach based on functional analysis leads to the distance of a perturbed eigenvalue from the whole true spectrum. In the approach by Koltchinskii and Giné (2000), the distance is measured with respect to a 2-norm between the sorted eigenvalues. Finally, our approach considers the maximum over the pair-wise errors of the first n approximate and true eigenvalues. Which kind of distance measure should be preferred certainly depends on the application, although the measure resulting from the functional analytic approach is rather weak.

The approximation bound depends on the norm of two error matrices which have been studied under fairly general assumptions in this chapter. For more specific settings, it might be possible to derive much more accurate error estimates, leading to improved bounds. Thus, the results as presented here do not form the conclusion to this approach, but there rather exists a well-defined interface for adopting these theoretical results to new applications: all that is required is providing accurate estimates of the error matrices.

We would like to restate the fact that the absolute factor is not merely an artifact, but rather reflects the structure of eigenvalues computed with finite precision arithmetics accurately. Since every matrix is already stored with a small perturbation, the eigenvalues have already been affected by a small additive perturbation, and a fully relative approximation error bound is not possible.

Obviously, most interesting are cases where the eigenvalues decay quickly. Otherwise, the approximation errors will not vary much between individual eigenvalues and there is no need to treat individual eigenvalues differently. For rapidly decaying eigenvalues, however, the truncation error

term also decays quickly and the truncation point r can be chosen such that the absolute error term becomes small.

Generally, it seems that the bounds are more interesting for fixed, or at least bounded r. In principle, it is possible to increase r depending on n, but depending on the type of kernel, the increase in r can be rather slow and the resulting error bound large. On the other hand, for fixed r, the relative error term in general decays as $O(n^{-\frac{1}{2}})$, leading to a typical stochastic convergence speed. Also note that there is no single r such that the bound is minimal for all eigenvalues at once. For small r, the bound tends to be smaller for large eigenvalues, where for large r, the bound becomes better for smaller eigenvalues.

Finally, it should also be stressed that the bounds are strong enough to prove that convergence of the eigenvalues takes place uniformly over all approximate eigenvalues.

3.13.2 Kernels with Bounded Eigenfunctions

The first class we have studied was that of a Mercer kernel whose eigenfunctions are uniformly bounded. An example was given by a kernel constructed using a sine basis. Sometimes, such a kernel is also referred to as a Fourier kernel.

For this setting, we have been able to derive a rather accurate finite sample size bound. In particular, it is possible to bound the truncation error $E(r,n)$ in a deterministic fashion. The relative error term $C(r,n)$ scales rather moderately as $r\sqrt{\log r}$ with r (see Section 3.11.1). In this case, it also turns out that assuming the best rate for adjusting r with respect to n, $E(r,n)$ decays quickly, depending on the rate of decay of the eigenvalues, for both the case of polynomial and exponential decay. Finally, if r is allowed to grow as $n \to \infty$, one obtains a bound which vanishes asymptotically. Its speed depends on the rate of decay of the eigenvalues. In the worst case, for polynomial eigenvalues which decay as $O(i^{-2})$, this rate is $O(n^{-\frac{1}{4}}\sqrt{\log n})$ while in the best case, the rate is $O(n^{-\frac{1}{2}}(\log n)^{\frac{3}{2}})$ which is only slightly slower than the (non-relative) absolute bounds.

All of these observations are also nicely reflected by the numerical simulations (see Figure 3.4), where the bounds decay as expected and provide good upper bounds on the approximation error. It should be stressed that in these cases, using an absolute error estimate would lead to an overestimation of the error for smaller eigenvalues by several order of magnitudes, although the absolute error bounds are asymptotically faster.

3.13.3 Bounded Kernel Functions

The second class of kernel functions were those which are uniformly bounded. This class includes the important radial basis function kernels (rbf-kernels). In this case, the eigenfunctions can in principle grow unboundedly as the eigenvalues become smaller, leading to considerably larger error estimates.

Still, as discussed in Section 3.11.2, the truncation error tends to 0 as $r \to \infty$, and the rate is slightly slower then in the case of bounded eigenfunctions. More importantly, the relative error term also depends on the eigenvalues themselves, and scales with the factor $1/\sqrt{\lambda_r}$. Therefore, having smaller eigenvalues can lead to a much larger relative error term (which will nevertheless ultimately decay to zero). This is also reflected in the achievable asymptotic rate, which is slower than in the case of bounded eigenfunctions.

In Section 3.12.2, we have discussed a concrete example, which has shown that there exist settings such that the relative error term actually scales as predicted with r. We conclude that in this case, convergence is actually much slower than in the case of bounded eigenfunctions.

The numerical simulations nevertheless reveal that the general structure of the bound matches the observed behavior, although the bounds should best be used for finite r, as explained above.

3.14 Conclusion

We have derived tight approximation bounds for the eigenvalues of the kernel matrix. These bounds correctly predict that smaller eigenvalues have much smaller approximation errors, as suggested by experimental evidence. The bounds consist of a relative term and an absolute error term, which is small if the eigenvalues decay quickly. We have shown that the absolute error term is not only an artifact of the derivation but actually reflects the structure of eigenvalues computed with finite-precision arithmetics. As such, the results presented here do not only apply to a purely theoretical setting, but already take care of finite-precision arithmetics, leading to estimates which hold for practical applications. We have considered two classes of kernel functions, kernels with uniformly bounded eigenfunctions and kernels which are uniformly bounded. These two examples address a number of relevant classes, especially those having an infinite expansion like rbf-kernels.

Chapter 4

Spectral Projections

Abstract

The subject of this chapter are the eigenvectors of the kernel matrix. We derive an upper bound, which scales with the corresponding eigenvalue, on the size of the scalar product between an eigenvector and the sample vector of a smooth function. This envelope supplements existing convergence results and shows that the coefficients of a smooth function with respect to the basis of eigenvectors of the kernel matrix decay as quickly as the corresponding eigenvalues.

4.1 Introduction

In the present chapter, we continue our investigation of the spectral properties of the kernel matrix. In the previous chapter, we have derived tight probabilistic finite-sample size bounds on the approximation error of the eigenvalues. The main focus has been laid on obtaining tight bounds for small eigenvalues. In this chapter, the eigenvectors of the kernel matrix will be studied. Formalizing convergence criteria for eigenvectors is a bit more involved, because for each sample size n, the eigenvectors lie in a different space, namely $\mathbb{R}^n$, while the asymptotic eigenfunctions are functions in some Banach space. Some procedure to compare these objects has to be devised. In our case, this will be another fixed function, the scalar product with which will form the test bed to compare eigenvectors with eigenfunctions. This setting is relevant because the first step in kernel ridge regression is precisely a scalar product between the eigenvectors of the kernel matrix and the label vector, which is equal to a sample vector of a smooth function plus noise.

General convergence of scalar products follows from a result on scalar projections (Koltchinskii, 1998). The goal of this chapter is not to improve upon this result, but to supplement the result by a tight envelope, which will again have a relative-absolute form. An *envelope*, in our terminology, is an upper bound which does not converge to zero as the number of samples tends to infinity. However, the envelope will show that certain approximation errors of the scalar products will be very small independent of the sample size.

For simplicity, let us assume that all eigenvalues have multiplicity one. Then, the above-mentioned result shows that the properly scaled scalar products of the sample vector of a fixed function f with the eigenvectors of the kernel matrix approximate the scalar products of the function f with the eigenfunctions of T_k. If f is one of the eigenfunction of ψ_i, this means that the estimated scalar products should converge to zero except for the ith eigenvector of the kernel matrix. The question is if the approximation error is the same for all eigenvectors of the kernel matrix, or not. In Figure 4.1, the estimated scalar products between a smooth function $f(x) = \mathrm{sinc}(4x)$, which is constructed from only a few eigenfunctions of T_k, is plotted. We see that the estimated scalar products decay rapidly which suggests that scalar products with eigenvectors of smaller eigenvalues will fluctuate less than those with eigenvectors of large eigenvalues. We are interested in computing an envelope such that the maximal approximation error for eigenvector u_i depends on the eigenvalue l_i.

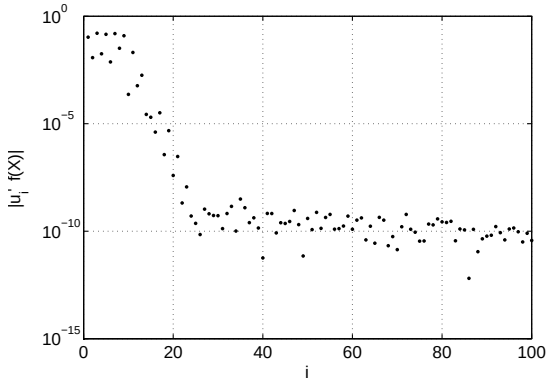


Figure 4.1: The absolute values of scalar products between the sample vector of $f(x) = \mathrm{sinc}(4x)$ and the eigenvectors of an rbf-kernel matrix decay quickly.

This chapter is structured as follows. Section 4.2 reviews the main results of this chapter in a less technical manner. Section 4.3 introduces some definitions used throughout this chapter. In Section 4.4, existing results on spectral projections are reviewed. The notion of a relative-absolute envelope is discussed in Section 4.5. Section 4.6 defines a road-map for the derivation of the main result by presenting the decomposition of the general case into three subproblems: (1) Scalar products between sample vectors of eigenfunctions and eigenvectors (Section 4.7), (2) eigenvector perturbations by truncation of the kernel function (Section 4.8), and (3) truncating general functions (Section 4.9). The main result is then derived in Section 4.10. Section 4.11 discusses the main results and Section 4.12 concludes this chapter.

4.2 Summary of Main Results

First, we show how an existing result on the convergence of spectral projections implies that scalar products with eigenvectors converge. However, the cited result does not contain finite sample size bounds with the desirable properties described in the introduction. We thus derive an envelope, which is an upper bound which becomes small for certain eigenvectors but which does not converge to 0 as the sample size goes to infinity.

For spectral projections, the basic setting consists of degenerate kernels and a single eigenfunction. In this situation, we will study scalar products of the form $|u_i^\top(\lambda_i\psi_i(\mathbf{X}))|/\sqrt{n}$ between eigenvectors of the kernel matrix and weighted eigenfunction sample vectors. Later, we will construct a result for a smooth function from these building blocks.

Theorem 4.1 (Relative Envelope for Degenerate Kernels)
(Theorem 4.34 in the main text) The scalar products between the eigenvectors u_i of the kernel matrix $\mathbf{K}$ for a degenerate kernel k and the sample vectors of eigenfunctions ψ_i of k multiplied by the corresponding eigenvalue are bounded as follows:

$$\frac{1}{\sqrt{n}}|\lambda_i\psi_i(\mathbf{X})^\top u_j| \leq l_j\|\mathbf{\Psi}^+\|, \tag{4.2}$$

where l_j is the jth eigenvalue of $\mathbf{K}$ and the columns of the matrix $\mathbf{\Psi}$ are given by sample vectors of the eigenfunctions ψ_i. As $n \to \infty$, $\|\mathbf{\Psi}^+\| \to 1$.

This shows that the sample vectors of the functions $\lambda_i\psi_i$ have a similar complexity as the original functions $\lambda_i\psi_i$. From this result, the full general relative-absolute envelope is derived which uses a similar technique as in the eigenvalue case to consider the full rank kernel matrix as an additive perturbation of the degenerate kernel matrix. The final result is:

Theorem 4.3 (Relative-Absolute Envelope for Scalar Products with Eigenvectors) *(Theorem 4.92 in the main text) Let u_i be the eigenvectors of the kernel matrix $\mathbf{K}$ based on an n-sample for a general Mercer kernel k, and let $f = \sum_{\ell=1}^{\infty} \alpha_\ell \lambda_\ell \psi_\ell$ be a function in the range of T_k, the integral operator induced by k. Then,*

$$\frac{1}{\sqrt{n}} |u_i^\top f(\mathbf{X})| \leq l_i C(r,n) + E(r,n) + T(r,n), \tag{4.4}$$

where l_i are the eigenvalues of the degenerate kernel matrix at rank r. The relative error term is

$$C(r,n) = O(\|\alpha^{[r]}\|_1 \|\mathbf{\Psi}^+\|), \tag{4.5}$$

where $\|\alpha^{[r]}\|_1$ is the 1-norm of the first r coefficients of $\alpha = (\alpha_\ell)$. The absolute error terms E and T are generated by the truncation of the kernel function and f and are small if r is chosen appropriately.

In other words, the size of the scalar products between the sample vector $f(\mathbf{X})$ and an eigenvector u_i of the kernel matrix is roughly bounded by a constant times the eigenvalue l_i. Therefore, with increasing i, the scalar products will decrease as quickly as the eigenvalues.

We argue that this result has a similar interpretation as the sampling theorem. A bandwidth limited function corresponds to a function with only finitely many non-zero coefficients in α. Then, for a finite sample, the empirical coefficients given by the scalar products with the eigenvectors of the kernel matrix also have only a finite number of approximately non-zero coefficients.

4.3 Preliminaries

As in the last chapter, we consider a Mercer kernel k on the Hilbert space $\mathcal{H}_\mu(\mathcal{X})$ (see Section 2.4). The integral operator associated with k is T_k (see (2.18)). The eigenvalues of T_k are $\lambda_1 \geq \lambda_2 \geq \ldots \to 0$, and the eigenfunctions are denoted by $(\psi_i)_{i \in \mathbb{N}}$. Recall that these form an orthogonal family of functions in $\mathcal{H}_\mu(\mathcal{X})$. We will also say that λ_i and ψ_i are the eigenvalues and eigenfunctions of k. These occur in Mercer's formula (2.16) defining the kernel function $k(x,y) = \sum_{i=1}^{\infty} \lambda_i \psi_i(x) \psi_i(y)$.

For a given r, the truncated kernel function is $k^{[r]}(x,y) = \sum_{i=1}^{r} \lambda_i \psi_i(x) \psi_i(y)$. The eigenvalues and eigenvectors of the kernel matrix $\mathbf{K}_n$ are l_i and u_i, and those of the truncated kernel matrix $\mathbf{K}_n^{[r]}$ are m_i and v_i.

4.4 Existing Results on Spectral Projections

Let us first review results on the convergence of spectral projections which also imply convergence for scalar products with eigenvectors. It is known that the eigenvectors of $\mathbf{K}_n$ converge to the eigenfunctions of T_k. This is again a consequence of the fact that $\mathbf{K}_n$ approximates the integral operator T_k in an appropriate sense. The following result is from Koltchinskii (1998). We introduce some definitions first.

Comparing eigenvectors is more complicated than comparing eigenvalues. The reason is that eigenvalues can correspond to eigenspaces with dimension larger than 1. Any vector from this eigenspace is a valid eigenvector. Therefore, simply comparing single eigenvectors against one another is not possible. Instead, one considers the projections to the given eigenspaces. These are invariant with respect to the choice of eigenvectors which span the eigenspaces. Furthermore, an eigenvalue with multiplicity larger than 1 becomes perturbed to a small cluster of eigenvalues. Therefore, for eigenvalues with multiplicity larger than 1, one has to compare the eigenprojection of a cluster of eigenvalues to the original eigenspace. In the result we will discuss below, a special kind of notion of a cluster is introduced which also includes a condition for well-separatedness from the rest of the spectrum.

Let K be a compact operator and denote with $\lambda(K)$ its spectrum. We will assume that K is positive semi-definite. The values of $\lambda(K)$ will as usual be numbered in non-increasing order $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$,

where the eigenvalues are repeated according to their multiplicity. Let $(i_j)_j$ be the subsequence such that i_j is the first occurrence of the eigenvalue λ_{i_j}.

An ε-*cluster* is a subset of $\lambda(K)$ with diameter less than ε whose distance from the remaining spectrum is larger than ε. For a fixed ε, we enumerate the ε-clusters in non-increasing order as $\Lambda_1^\varepsilon(K), \Lambda_2^\varepsilon(K), \ldots$. Note that for a given ε, it is possible that no such cluster exists.

Define the minimum separation of the first r distinct eigenvalues from the remaining spectrum:

$$\delta_r(K) = \min_{1\le j\le r} d(\lambda_{i_j}, \lambda(K) \setminus \{\lambda_{i_j}\}). \tag{4.6}$$

Now, if $\varepsilon < \delta_r(K)$, then the first r ε-clusters exist, and $\Lambda_j^\varepsilon(K) = \{\lambda_{i_j}\}$ for $1 \le j \le r$.

Furthermore, denote by $\pi_j(K)$ the orthogonal projection onto the eigenspace of the eigenvalue λ_{i_j}, and let $\pi_j^\varepsilon(K)$ be the projection onto the eigenspace belonging to the jth ε-clusters $\Lambda_j^\varepsilon(K)$. Again, if $\varepsilon < \delta_r(K)$, then $\pi_j(K) = \pi_j^\varepsilon(K)$ for $1 \le j \le r$.

The convergence result will be formulated in terms of the projections of the rth ε-cluster of $\mathbf{K}_n$, and the rth eigenprojection of T_k (Defining the expressions for a matrix $\mathbf{K}_n$ analogously). Since the projections $\pi_r^\varepsilon(\mathbf{K}_n)$ and $\pi_r(T_k)$ operate on different spaces ($\mathbb{R}^n$ and the Hilbert space $\mathcal{H}_\mu(\mathcal{X})$), instead, the bilinear mappings induced by these projections are considered. These are:

$$\beta_r^n(f,g) = \langle \pi_r^\varepsilon(\mathbf{K}_n) f, g\rangle_{\mu_n} \quad \text{and} \quad \beta_r(f,g) = \langle \pi_r(T_k) f, g\rangle_\mu. \tag{4.7}$$

For greater clarity, let us spell out β_r^n. Let $u_1, \ldots, u_d$ be an orthonormal basis of the eigenspace of $\Lambda_r^\varepsilon(\mathbf{K}_n)$, Then,

$$\beta_r^n(f,g) = \frac{1}{n}\big(\pi_r^\varepsilon(\mathbf{K}_n) f(\mathbf{X})\big)^\top g(\mathbf{X}) = \frac{1}{n}\sum_{j=1}^{d}\big(u_j u_j^\top f(\mathbf{X})\big)^\top g(\mathbf{X}). \tag{4.8}$$

The convergence result states convergence of $\beta_r^n \to \beta_r$ uniformly over a set of functions $\mathcal{F}$ which have a finite complexity in the following sense. Call $\mathcal{F}$ μ-Glivenko-Cantelli (or $\mathcal{F} \in GC(\mu)$), if

$$\sup_{f\in\mathcal{F}} |\mathbf{E}_{\mu_n} f - \mathbf{E}_\mu f| \to_{\text{a.s.}} 0. \tag{4.9}$$

Finally, we come to the strong law of large numbers which is part of the theorem of (Koltchinskii, 1998, Theorem 2.1).

Theorem 4.10 *Let k be a Mercer kernel on $\mathcal{H}_\mu(\mathcal{X})$ with eigenvalues $(\lambda_i)_{i\in\mathbb{N}}$ and eigenfunctions $(\psi_i)_{i\in\mathbb{N}}$. Let $\mathcal{F}$ be a class of measurable functions on $\mathcal{X}$ with a square integrable majorant F in $\mathcal{H}_\mu(\mathcal{X})$, such that for each $i \in \mathbb{N}$,*

$$\mathcal{F}\psi_i := \{f\psi_i \colon f \in \mathcal{F}\} \in GC(\mu). \tag{4.11}$$

Then, for all $r \in \mathbb{N}$ and $\varepsilon < \delta_{r/4}(T_k)$, $\beta_r^\varepsilon \to \beta_r$ uniformly over $\mathcal{F}$:

$$\sup_{f,g\in\mathcal{F}} \Big| \langle \pi_r^\varepsilon(\mathbf{K}_n) f, g\rangle_{\mu_n} - \langle \pi_r(T_k) f, g\rangle_\mu \Big| \to_{\text{a.s.}} 0. \tag{4.12}$$

In words, for sufficiently small ε, eventually, $\pi_r^\varepsilon(\mathbf{K}_n)$ (which need not exist for larger ε, since $\delta_r(T_k)$ is defined with respect to the eigenvalues of T_k, not $\mathbf{K}_n$) begins to capture a cluster of eigenvalues which form an approximation of the eigenvalue λ_{i_r}. Moreover, the associated eigenspace converges such that the projection together with the approximation of the scalar product converges to the true projection and the true scalar product.

Also note that the result is uniform over $\mathcal{F}$, but only holds for individual distinct eigenvalues at a time. Therefore, the result is trivially extensible only to a finite number of scalar products.

Theorem 4.10 also implies a result on scalar products between eigenvectors of $\mathbf{K}_n$ and eigenfunctions of T_k. Let ψ be an eigenfunction of T_k, set $\mathcal{F} = \{\psi\}$, and let $u_1, \ldots, u_d$ be an orthonormal basis of the eigenspace of $\Lambda_r^{\varepsilon}(\mathbf{K}_n)$). Then,

$$\begin{aligned}\langle \pi_r^{\varepsilon}(\mathbf{K}_n)\psi, \psi\rangle_{\mu_n} &= \frac{1}{n}\sum_{i=1}^{d}\big(u_i u_i^{\top}\psi(\mathbf{X})\big)^{\top}\psi(\mathbf{X}) \\ &= \frac{1}{n}\sum_{i=1}^{d}(u_i^{\top}\psi(\mathbf{X}))^{\top}(u_i^{\top}\psi(\mathbf{X})) = \frac{1}{n}\sum_{i=1}^{d}(u_i^{\top}\psi(\mathbf{X}))^2\end{aligned} \tag{4.13}$$

On the other hand,

$$\langle \pi_r(T_k)\psi, \psi\rangle_{\mu} = \begin{cases}1 & \psi \text{ is eigenfunction to eigenvalue } \lambda_{i_r}, \\ 0 & \text{else,}\end{cases} \tag{4.14}$$

because the eigenspaces are orthogonal. Therefore,

$$\frac{1}{n}\sum_{i=1}^{d}(u_i^{\top}\psi(\mathbf{X}))^2 \to \begin{cases}1 & \psi \text{ is eigenfunction to eigenvalue } \lambda_{i_r}, \\ 0 & \text{else.}\end{cases} \tag{4.15}$$

If all eigenvalues have multiplicity 1, this reduces to

$$\frac{1}{\sqrt{n}}|u_i^{\top}\psi_j(\mathbf{X})| \to \delta_{ij}, \tag{4.16}$$

where $u_1, \ldots, u_n$ are the eigenvectors of $\mathbf{K}_n$ and $(\psi_i)_{i\in\mathbb{N}}$ are the eigenfunctions of T_k. Therefore, the empirical scalar products converge to the true scalar products.

An alternative approach for proving convergence of the eigenvectors is based on functional analysis and has for example been followed in the Ph.D. thesis of von Luxburg (2004). There exist fairly general theorems relating the distance of eigenvalues to closeness of two operators which may live on infinite-dimensional function spaces. These results also hold for non-self-adjoint operators. The crucial step is to interpret the kernel matrix $\mathbf{K}_n$ as an operator K_n on $C(\mathcal{X})$ via

$$K_n f(x) = \frac{1}{n}\sum_{i=1}^{n} k(x, X_i) f(X_i), \tag{4.17}$$

which has the same eigenvalues (as an operator on $C(\mathcal{X})$) as $\mathbf{K}_n$ (as an operator on $\mathbb{R}^n$), and to show that $K_n \to T_k$ in some appropriate sense. This step usually relies on certain regularity properties of the kernel function, for example, equicontinuity. After convergence has been established, convergence of the eigenvalues and eigenvectors follow by standard arguments, which can for example be found in Anselone (1971) or Engl (1997).

A weakness of this approach is that due to the generality of the setting, it becomes hard to exploit useful properties effectively, for example self-adjointness of the operator which is implied by symmetry of the kernel function. As a result, the results are considerably less accurate compared with the approach discussed above. For example, convergence of eigenvalues is formulated in the sense that eventually, an eigenvalue of K_n will lie in any neighborhood of the spectrum of T_k. This is a weaker statement than the pairwise uniform convergence of the ith approximate eigenvalue to the true ith eigenvalue. This also excludes the kind of accurate error estimate as given in the previous chapter.

With respect to eigenvectors, convergence is shown in that there exists a subsequence of the eigenvectors of K_n which converge to an eigenvector of T_k.

In principle, it should be possible to derive finite sample size bounds using this functional analytic approach, but this process seems to be rather tedious while at the same time not leading to sufficiently accurate estimates. The reason is again that the bounds hold for fairly general operators and fail to exploit specific properties of more well-behaved classes of operators.

4.5 A Relative-Absolute Envelope for Scalar Products

The result cited in the last section establishes quite generally that spectral projections and scalar products converge. Again, we are looking into a special situation for which we want to obtain tight finite sample size bounds on the individual approximation error.

The situation we are interested in is the scalar product between the eigenvectors of $\mathbf{K}_n$ and a smooth function f sampled at $X_1, \dots, X_n$. In our context, *smooth* means that the function lies in the image of T_k. There are several ways to see why T_k generates functions with bounded complexity. For example, T_k is a compact operator, which is an operator which maps a bounded set to a compact set. A compact set of functions has finite complexity because it allows a finite covering with spheres of any non-zero size. This means that at a given scale, there seem to be only a finite number of different functions in that set. Now if $f = T_k g$, then $f \in T_k \mathcal{B}(\|g\|)$, where $\mathcal{B}(\alpha) = \{f \in \mathcal{H} \mid \|f\| \leq \alpha\}$. Therefore, f lies in a compact set which has finite complexity.

Another reason is that f inherits regularity parameters like the Lipschitz-constant or a bounded derivative from the kernel function k depending on the norm of g. The larger g, the more amplified are the regularity parameters of k, leading to more irregular functions.

A consequence of these observations is that the eigenfunctions of the kernel become more irregular the smaller the eigenvalue is. If ψ is an eigenfunction to eigenvalue λ, then

$$\psi = \frac{1}{\lambda} T_k \psi. \tag{4.18}$$

Therefore, ψ is the image under T_k of ψ/λ. The smaller λ is, the larger is the norm of ψ/λ, and the more irregular is ψ.

Now if $f = T_k g$, we can write f using the spectral decomposition of T_k as

$$f = \sum_{i=1}^{\infty} \alpha_i \lambda_i \psi_i, \qquad \text{with } \alpha_i = \langle g, \psi_i \rangle, \text{ and } (\alpha_i) \in \ell^2. \tag{4.19}$$

The $\lambda_i \alpha_i$ constitute the scalar products between f and the eigenfunctions of T_k. Because $(\lambda_i \alpha_i)$ is the component-wise product of the ℓ^2-sequence (α_i) and the ℓ^1-sequence (λ_i), the scalar products $\langle f, \psi_i \rangle$ decay fairly quickly for a smooth function.

We are interested in the question whether this also holds for the empirical scalar products $\frac{1}{\sqrt{n}} f(\mathbf{X})^\top u_i$, with $f(\mathbf{X}) = (f(X_1), \dots, f(X_n))^\top$. From the results cited in the last section, we only know that the individual scalar products converge, also uniformly over GC-families, but not how the error is distributed across the individual eigenspaces. In particular, we are interested whether the empirical scalar products also decay quickly. This means that besides general convergence, the coefficients of f with respect to the eigenfunctions of T_k and the eigenvectors of $\mathbf{K}_n$ share an important structural property: the coefficients decay quickly with decreasing eigenvalue.

The goal of the remainder of this chapter is devoted to deriving an envelope complementing the general convergence result. This envelope bounds the absolute value of the scalar product $\frac{1}{\sqrt{n}} f(\mathbf{X})^\top u_i$ for varying i. This will be done by considering the individual functions $\lambda_i \psi_i(\mathbf{X})$ from which f is constructed. For these functions, we are particularly interested in how large

$$u_i^\top (\lambda_j \psi_j(\mathbf{X})) \tag{4.20}$$

is if $i > j$. If the scalar product of $\lambda_j \psi_j(\mathbf{X})$ with u_i decays fairly quickly after $i > j$, this also means that the coefficient of any f will decay as quickly as does α_i.

4.6 Decomposition of the General Case

We begin by deriving the basic decomposition of the general case which identifies the subproblems to be treated in the subsequent sections. Thus, this sections serves as a rough road-map of the derivation of the main result. The general case is given by considering the scalar products between the eigenvectors $u_1, \dots, u_n$ of the kernel matrix $\mathbf{K}_n$, and a function f which obeys (4.19).

We will use two truncation steps to reduce the general problem to finite-dimensional problems. Given a general Mercer kernel k as in Chapter 3, we have defined its r-truncation as

$$k^{[r]}(x,y) = \sum_{i=1}^{r} \lambda_i \psi_i(x)\psi_i(y). \tag{4.21}$$

Accordingly, the kernel matrix based on $k^{[r]}$ will be denoted by $\mathbf{K}^{[r]}$. The resulting *truncation error matrix* $\mathbf{E}^r$ is defined by

$$\mathbf{E}^r = \mathbf{K} - \mathbf{K}^{[r]}, \tag{4.22}$$

where, in contrast to Chapter 3, we have omitted the index n for the sake of simplicity.

The other truncation will be applied to the function f as defined in (4.19). Its r-truncation is given by

$$f^{[r]} = \sum_{\ell=1}^{r} \lambda_\ell \alpha_\ell \psi_\ell. \tag{4.23}$$

Note that the image of $\mathbf{K}^{[r]}$ is spanned by the vectors $\psi_1(\mathbf{X}), \ldots, \psi_r(\mathbf{X})$, which implies that $f^{[r]}(\mathbf{X}) \in \operatorname{ran} \mathbf{K}^{[r]}$.

The decomposition is carried out in three steps:

1. *Truncation of f.* By substituting $f = f^{[r]} + f - f^{[r]}$, we obtain that

$$\begin{aligned} \frac{1}{\sqrt{n}}|u_i^\top f(\mathbf{X})| &\leq \frac{1}{\sqrt{n}}|u_i^\top f^{[r]}(\mathbf{X})| + \frac{1}{\sqrt{n}}|u_i^\top f(\mathbf{X}) - u_i^\top f^{[r]}(\mathbf{X})| \\ &\leq \frac{1}{\sqrt{n}}|u_i^\top f^{[r]}(\mathbf{X})| + \|u_i\| \frac{1}{\sqrt{n}}\|f(\mathbf{X}) - f^{[r]}(\mathbf{X})\|, \end{aligned} \tag{4.24}$$

   by the Cauchy-Schwarz inequality. Note that $\|u_i\| = 1$. The first term is now the scalar product between the ith eigenvector of $\mathbf{K}$ and a function which lies in the span of the sample vectors of the first r eigenfunctions of k.

2. *Introducing eigenvectors of $\mathbf{K}^{[r]}$.* Since $f^{[r]}(\mathbf{X}) \in \operatorname{ran} \mathbf{K}^{[r]}$, and the image of $\mathbf{K}^{[r]}$ is spanned by the first r eigenvectors $v_1, \ldots, v_r$ of $\mathbf{K}^{[r]}$, it holds that

$$f^{[r]}(\mathbf{X}) = \sum_{j=1}^{r} v_j v_j^\top f^{[r]}(\mathbf{X}). \tag{4.25}$$

   Therefore,

$$u_i^\top f^{[r]}(\mathbf{X}) = \sum_{j=1}^{r} (u_i^\top v_j)(v_j^\top f^{[r]}(\mathbf{X})). \tag{4.26}$$

3. *Expanding $f^{[r]}$.* Finally, we plug in the expansion of $f^{[r]}$ in terms of the (finitely many) terms $\alpha_\ell \lambda_\ell \psi_\ell$, such that

$$v_j^\top f^{[r]}(\mathbf{X}) = \sum_{\ell=1}^{r} \alpha_\ell (\lambda_\ell \psi_\ell(\mathbf{X})^\top v_j). \tag{4.27}$$

Let us summarize the resulting decomposition in the following lemma:

Lemma 4.28 *The scaled scalar product between u_i and the sample vector $f(\mathbf{X})$ can be upper bounded as follows:*

$$\frac{1}{\sqrt{n}}|u_i^\top f(\mathbf{X})| \leq \sum_{\ell=1}^{r} |\alpha_\ell| \sum_{j=1}^{r} |u_i^\top v_j| \frac{1}{\sqrt{n}}|\lambda_\ell \psi_\ell(\mathbf{X})^\top v_j| + \frac{1}{\sqrt{n}}\|f(\mathbf{X}) - f^{[r]}(\mathbf{X})\|. \tag{4.29}$$

In the remainder of this chapter, we will first treat the individual components of the derivation before we finally state the closing main result. These elements are:

- Scalar products between sample vectors of eigenfunctions and eigenvectors of degenerate kernels:
$$\frac{1}{\sqrt{n}}|\lambda_\ell \psi_\ell(\mathbf{X})^\top v_j|. \tag{4.30}$$
- The relation between the eigenvectors of $\mathbf{K}$ and $\mathbf{K}^{[r]}$:
$$|u_i^\top v_j|. \tag{4.31}$$
- The truncation error for f:
$$\frac{1}{\sqrt{n}}\|f(\mathbf{X}) - f^{[r]}(\mathbf{X})\|. \tag{4.32}$$

4.7 Degenerate Kernels and Eigenfunctions

The first step treats scalar products between degenerate kernels and sample vectors of eigenfunctions. Therefore, let k be an r-degenerate kernel with eigenvalues $\lambda_1, \ldots, \lambda_r$ and eigenfunctions $\psi_1, \ldots, \psi_r$ on $\mathcal{H}_\mu(\mathcal{X})$. Later on, this degenerate kernel will be obtained by truncating a general kernel. For the sake of notational simplicity, we will write k in this section, assuming that k is already degenerate. Also, we will suppress the n in the index since the computations here are mostly algebraic, in contrast to Chapter 3, where asymptotic analyses were also performed.

Analogously to the last chapter we define the $n \times r$ matrix $\mathbf{\Psi}$ having the entries

$$[\mathbf{\Psi}]_{i\ell} = \frac{1}{\sqrt{n}}\psi_\ell(X_i), \tag{4.33}$$

where $X_1, \ldots, X_n$ are i.i.d. samples in $\mathcal{X}$ distributed according to μ. As shown in (3.69), it holds that the (normalized) kernel matrix $\mathbf{K} = (\frac{1}{n}k(X_i, X_j))_{i,j=1}^n$ is $\mathbf{K} = \mathbf{\Psi}\mathbf{\Lambda}\mathbf{\Psi}^\top$, where $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_r)$.

Theorem 4.34 *Let $\mathbf{K}$ be the normalized kernel matrix for a degenerate kernel k based on an n-sample $X_1, \ldots, X_n$. Furthermore let $\mathbf{K} = \mathbf{V}\mathbf{M}\mathbf{V}^\top$ be the spectral decomposition of $\mathbf{K}$. Let $v_1, \ldots, v_n$ be the columns of $\mathbf{V}$ and $m_1, \ldots, m_n$ the diagonal elements of $\mathbf{M}$. Then, if $\mathbf{\Psi}^\top\mathbf{\Psi}$ is invertible, it holds for $1 \le j \le n$ that*

$$\|\mathbf{\Lambda}\mathbf{\Psi}^\top v_j\| \le m_j\|\mathbf{\Psi}^+\|, \tag{4.35}$$

where $\mathbf{\Psi}^+$ denotes the pseudo-inverse of $\mathbf{\Psi}$. For all $1 \le \ell \le r$,

$$\frac{1}{\sqrt{n}}|\lambda_\ell \psi_\ell(\mathbf{X})^\top v_j| \le m_j\|\mathbf{\Psi}^+\|. \tag{4.36}$$

Furthermore, if $n > r$, $v_{r+1}, \ldots, v_n$ span a subspace of the nullspace of $\mathbf{K}$, and

$$\mathbf{\Psi}^\top v_j = 0 \tag{4.37}$$

for $r + 1 \le j \le n$.

Proof Since $\mathbf{K} = \mathbf{\Psi}\mathbf{\Lambda}\mathbf{\Psi}^\top$,

$$\begin{aligned} & \mathbf{\Psi}\mathbf{\Lambda}\mathbf{\Psi}^\top v_j = m_j v_j \\ [\mathbf{\Psi}^\top\times] \quad & \mathbf{\Psi}^\top\mathbf{\Psi}\mathbf{\Lambda}\mathbf{\Psi}^\top v_j = m_j\mathbf{\Psi}^\top v_j \\ [(\mathbf{\Psi}^\top\mathbf{\Psi})^{-1}\times] \quad & \mathbf{\Lambda}\mathbf{\Psi}^\top v_j = m_j(\mathbf{\Psi}^\top\mathbf{\Psi})^{-1}\mathbf{\Psi}^\top v_j = m_j\mathbf{\Psi}^+ v_j. \end{aligned} \tag{4.38}$$

Taking norms on both sides gives

$$\|\mathbf{\Lambda}\mathbf{\Psi}^\top v_j\| \leq m_j \|\mathbf{\Psi}^+\|. \tag{4.39}$$

Now since trivially, the 2-norm bounds each individual component of a vector, and $[\mathbf{\Lambda}\mathbf{\Psi}^\top v_j]_\ell = \lambda_\ell \psi_\ell(\mathbf{X})^\top v_j/\sqrt{n}$ (because $[\mathbf{\Lambda}\mathbf{\Psi}^\top]_{\ell i} = \lambda_\ell [\mathbf{\Psi}]_{i\ell} = \lambda_\ell \psi_\ell(X_i)/\sqrt{n}$),

$$\frac{1}{\sqrt{n}}|\lambda_\ell \psi_\ell(\mathbf{X})^\top v_j| \leq m_j \|\mathbf{\Psi}^+\|. \tag{4.40}$$

The kernel matrix $\mathbf{K}$ has at most rank r. Therefore, if $r < n$, and the columns of $\mathbf{V}$ are sorted in non-increasing order, $m_{r+1}, \ldots, m_n$, and $v_{r+1}, \ldots, v_n$ lie in the nullspace of $\mathbf{K}$, and are orthogonal to the image of $\mathbf{K}$ which lies inside $v_1, \ldots, v_r$. On the other hand, the image of $\mathbf{K}$ is also spanned by the columns of $\mathbf{\Psi}$. Therefore, $\mathbf{\Psi}^\top v_j = 0$ for $r+1 \leq j \leq n$. ■

We conclude this section by relating the size of the pseudo-inverse to $\|\mathbf{\Psi}^\top\mathbf{\Psi} - \mathbf{I}\|$, which was called the *relative error term* $\|\mathbf{C}\|$ in Chapter 3.

Recall that the norm of the pseudo inverse $\mathbf{\Psi}^+$ is the inverse of the smallest singular value of $\mathbf{\Psi}$:

$$\|\mathbf{\Psi}^+\| = 1/\sigma_n(\mathbf{\Psi}). \tag{4.41}$$

The singular values are the square roots of the eigenvalues of $\mathbf{\Psi}^\top\mathbf{\Psi}$. First of all, note that $\|\mathbf{\Psi}^\top\mathbf{\Psi} - \mathbf{I}\| = \max_i |\lambda_i(\mathbf{\Psi}^\top\mathbf{\Psi}) - 1|$, and therefore $\|\mathbf{\Psi}^\top\mathbf{\Psi} - \mathbf{I}\| \to 0$ implies that $\lambda_i(\mathbf{\Psi}^\top\mathbf{\Psi}) \to 1$ for all $1 \leq i \leq n$. Furthermore,

$$1 - \lambda_n(\mathbf{\Psi}^\top\mathbf{\Psi}) \leq \max_{1\leq i\leq n} |\lambda_i(\mathbf{\Psi}^\top\mathbf{\Psi}) - 1| \leq \|\mathbf{\Psi}^\top\mathbf{\Psi} - \mathbf{I}\| \quad \Rightarrow \quad \lambda_n(\mathbf{\Psi}^\top\mathbf{\Psi}) \geq 1 - \|\mathbf{\Psi}^\top\mathbf{\Psi} - \mathbf{I}\|. \tag{4.42}$$

Therefore, $\sigma_n(\mathbf{\Psi}) = \sqrt{\lambda_n(\mathbf{\Psi}^\top\mathbf{\Psi})} \geq \sqrt{1 - \|\mathbf{\Psi}^\top\mathbf{\Psi} - \mathbf{I}\|}$, and it follows that

$$\|\mathbf{\Psi}^+\| = \frac{1}{\sigma_n(\mathbf{\Psi})} \leq \frac{1}{\sqrt{1 - \|\mathbf{\Psi}^\top\mathbf{\Psi} - \mathbf{I}\|}}. \tag{4.43}$$

We have proven the following lemma:

Lemma 4.44 *Under the conditions of the previous theorem, it holds that*

$$\|\mathbf{\Psi}^+\| \leq \frac{1}{\sqrt{1 - \|\mathbf{\Psi}^\top\mathbf{\Psi} - \mathbf{I}\|}}. \tag{4.45}$$

Thus, since $\|\mathbf{\Psi}^\top\mathbf{\Psi} - \mathbf{I}\| \to 0$ *almost surely, it follows that* $\|\mathbf{\Psi}^+\| \to 1$.

4.8 Eigenvector Perturbations for General Kernel Functions

The next step consists in relating the scalar products between sample vectors of eigenfunctions and the eigenvectors of the degenerate kernels $\mathbf{K}^{[r]}$. In Lemma 4.28, we have seen that this is accomplished by multiplying these scalar products with the scalar products between the eigenvectors of $\mathbf{K}$ and $\mathbf{K}^{[r]}$:

$$\frac{1}{\sqrt{n}}|\lambda_\ell \psi_\ell(\mathbf{X})^\top u_j| \leq \sum_{j=1}^{r} |u_i^\top v_j| \frac{1}{\sqrt{n}}|\lambda_\ell \psi_\ell(\mathbf{X})^\top v_j|. \tag{4.46}$$

In Theorem 4.34, we have proved that

$$\frac{1}{\sqrt{n}}|\lambda_\ell \psi_\ell(\mathbf{X})^\top v_j| \leq m_j \|\mathbf{\Psi}^+\|. \tag{4.47}$$

Therefore,

$$\sum_{j=1}^{r} |u_i^\top v_j| \frac{1}{\sqrt{n}}|\lambda_\ell \psi_\ell(\mathbf{X})^\top v_j| \leq \|\mathbf{\Psi}^+\| \sum_{j=1}^{r} |u_i^\top v_j| m_j. \tag{4.48}$$

The last sum is the expression we will study in this section.

Recall that $u_1, \ldots, u_n$ are the eigenvectors of $\mathbf{K}$ and $v_1, \ldots, v_r$ are those of $\mathbf{K}^{[r]}$. We interpret $\mathbf{K}$ as being an additive perturbation of $\mathbf{K}^{[r]}$, $\mathbf{K} = \mathbf{K}^{[r]} + \mathbf{E}^r$. The vector

$$s = \left(u_i^\top v_1, \ldots, u_i^\top v_r\right). \tag{4.49}$$

contains the coefficients of u_i with respect to the eigenbasis of $\mathbf{K}^{[r]}$. Therefore, these scalar products measures the perturbation of v_i to u_i induced by the additive perturbation $\mathbf{E}^r$. If $\|\mathbf{E}^r\| = 0$, $u_i = v_i$, and since the v_j are orthogonal, only $[s]_i = 1$, with all other entries being zero. For non-zero perturbations, u_i will be perturbed away from v_i leading to a spreading of the coefficients away from the configuration of all coefficients being zero except for $[s]_i = 1$. We wish to study the amount of this perturbation, and the effect this has on the sum $\sum_{j=1}^r |u_i^\top v_j| m_j$.

The first question is addressed by a family of general results on perturbation of eigenvectors, known as *sin-theta-theorems*.

The following Lemma is a special case of (Davis and Kahan, 1970, Theorem 6.2) (see also (Eisenstat and Ipsen, 1994; Stewart and Sun, 1990))

Lemma 4.50 *Let $\mathbf{A}$ be a symmetric $n \times n$ matrix with spectral decomposition $\mathbf{U}\mathbf{L}\mathbf{U}^\top$. Let $\mathbf{U}$ and $\mathbf{L}$ be partitioned as follows.*

$$\mathbf{U} = [\mathbf{U}_1\ \mathbf{U}_2], \qquad \mathbf{L} = \begin{bmatrix} \mathbf{L}_1 & 0 \\ 0 & \mathbf{L}_2 \end{bmatrix}, \tag{4.51}$$

where $\mathbf{U}_1 \in \mathbb{M}_{n,k}$, $\mathbf{L}_1 \in \mathbb{M}_k$, $\mathbf{U}_2 \in \mathbb{M}_{n,n-k}$, and $\mathbf{L}_2 \in \mathbb{M}_{n-k}$. Furthermore, let $\mathbf{E}$ be another symmetric matrix and $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{E}$. Let $\tilde{l}$ be an eigenvalue of $\tilde{\mathbf{A}}$ and $\tilde{x}$ an associated unit-length eigenvector. Then,

$$\|\mathbf{U}_2^\top \tilde{x}\| \leq \frac{\|\mathbf{E}\|}{\min_{n-k \leq i \leq n} |\tilde{l} - l_i|}. \tag{4.52}$$

Proof It holds that

$$(\mathbf{A} + \mathbf{E})\tilde{x} = \tilde{l}x \quad \Rightarrow \quad \mathbf{E}\tilde{x} = (\tilde{l}\mathbf{I} - \mathbf{A})\tilde{x}. \tag{4.53}$$

Therefore,

$$\|\mathbf{E}\| \geq \|\mathbf{E}\tilde{x}\| = \|(\tilde{l}\mathbf{I} - \mathbf{A})\tilde{x}\| = \|(\tilde{l}\mathbf{I} - \mathbf{U}\mathbf{L}\mathbf{U}^\top)\tilde{x}\| \tag{4.54}$$

This norm becomes smaller when we only consider the last $n - k$ components of the resulting vector. This part is computed by $(\tilde{l}\mathbf{I} - \mathbf{U}_2\mathbf{L}_2\mathbf{U}_2^\top)\tilde{x}$. Therefore, we continue (4.54):

$$\|\mathbf{E}\tilde{x}\| \geq \|(\tilde{l}\mathbf{I} - \mathbf{U}_2\mathbf{L}_2\mathbf{U}_2^\top)\tilde{x}\| = \|\mathbf{U}_2(\tilde{l}\mathbf{I} - \mathbf{L}_2)\mathbf{U}_2^\top \tilde{x}\|, \tag{4.55}$$

because $\mathbf{U}_2\mathbf{U}_2^\top = \mathbf{I}$. Finally,

$$\|\mathbf{U}_2(\tilde{l}\mathbf{I} - \mathbf{L}_2)\mathbf{U}_2^\top \tilde{x}\| = \|(\tilde{l}\mathbf{I} - \mathbf{L}_2)\mathbf{U}_2^\top \tilde{x}\| \geq \min_{n-k \leq i \leq n} |\tilde{l} - l_i| \|\mathbf{U}_2^\top \tilde{x}\|. \tag{4.56}$$

Dividing by $\min_i |\tilde{l} - l_i|$ concludes the proof of the lemma. ■

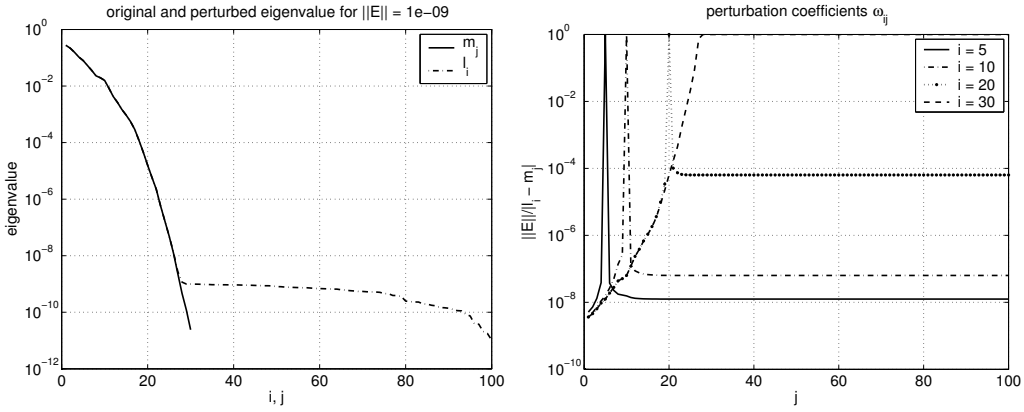
This lemma has a simple corollary for the case where one considers scalar products between individual eigenvectors of $\mathbf{A}$ and $\tilde{\mathbf{A}}$:

Corollary 4.57 *Denote the eigenvalues of $\mathbf{K}$ by l_i and those of $\mathbf{K}^{[r]}$ by m_j. Let the corresponding eigenvectors be u_i, and v_j respectively. Then,*

$$|u_i^\top v_j| \leq \frac{\|\mathbf{E}^r\|}{|l_i - m_j|} \wedge 1 =: \omega_{ij} \tag{4.58}$$

where $a \wedge b = \min(a, b)$. The numbers ω_{ij} will be called perturbation coefficients.

Proof The corollary follows from the previous lemma by setting $\mathbf{A} = \mathbf{K}^{[r]}$, $\tilde{\mathbf{A}} = \mathbf{K}$, $\mathbf{E} = \mathbf{E}^r$, and setting $\mathbf{U}_2$ equal to a $n \times 1$ matrix equal to v_j. The scalar product cannot become larger than 1 because $|u_i^\top v_j| \leq \|u_i\| \|v_j\| = 1$, and u_i, v_j are unit length vectors. ■



(a) The original and perturbed eigenvalues. (b) The resulting perturbation coefficients.

Figure 4.2: Example plots for perturbation coefficients ω_{ij}. If $\|\mathbf{E}^r\|$ is small and the eigenvalues decay quickly, the large eigenvalues are well-separated such that the perturbation of the eigenvectors is negligibly small.

This is a classical result which is usually paraphrased as the perturbation being small if the eigenvalues are well-separated. In our case, we assume that the eigenvalues decay to zero, such that the eigenvalue become clustered around 0 and seem anything but well-separated. However, note that the separation is measured at the scale of $\|\mathbf{E}^r\|$. In Chapter 3, we have seen that $\|\mathbf{E}^r\| \to 0$ as r increases, such that $\|\mathbf{E}^r\|$ will be rather small typically, and eigenvalues which are close together can be well-separated nevertheless. Now, for $|l_i - m_j| > \|\mathbf{E}^r\|$ we can re-write

$$\omega_{ij} = \frac{1}{\frac{|l_i - m_j|}{\|\mathbf{E}^r\|}}. \tag{4.59}$$

Typically, $j \mapsto \omega_{ij}$ will have the following shape for fixed i (see Figure 4.2). In Figure 4.2(b), each line describes the characteristics of the perturbation of a single eigenvector. The perturbation coefficient ω_{ij} will be 1 for eigenvalues m_j which are closer than $\|\mathbf{E}^r\|$ to l_i. For larger eigenvalues, ω_{ij} drops off fairly quickly, as it does for smaller eigenvalues, although it eventually starts to reach a plateau and not decay further. Roughly stated, if l_i is still much larger than $\|\mathbf{E}^r\|$, and l_i is isolated, then ω_{ij} will have a single peak of 1 at ω_{ii} and be negligibly small for $\omega_{i1}, \ldots, \omega_{i,j-1}, \omega_{i,j+1}, \ldots, \omega_{in}$. For small eigenvalues l_i, the perturbation can be rather severe, although we see that the perturbation will occur mostly in the direction of eigenspaces to comparably small eigenvalues.

Now, we return to the sum

$$\sum_{j=1}^{r} \omega_{ij} m_j. \tag{4.60}$$

We will show that outside of a relatively small set around l_i, ω_{ij} will be of the order of $\|\mathbf{E}^r\|$.

Lemma 4.61 *Consider*

$$\omega_{ij} m_j = \left(\frac{\|\mathbf{E}^r\|}{|l_i - m_j|} \wedge 1 \right) m_j. \tag{4.62}$$

Then,

$$m_j \geq 2l_i \qquad \Rightarrow \qquad \omega_{ij} m_j \leq 2\|\mathbf{E}^r\|, \tag{4.63}$$

$$m_j \leq \frac{1}{2} l_i \qquad \Rightarrow \qquad \omega_{ij} m_j \leq \|\mathbf{E}^r\|. \tag{4.64}$$

Proof For this proof, we will drop the superscript r on $\mathbf{E}^r$ for convenience. First, note that for $m_j = 2l_i$,

$$\frac{\|\mathbf{E}\|m_j}{|l_i - m_j|} = \frac{2\|\mathbf{E}\|l_i}{2l_i - l_i} = 2\|\mathbf{E}\|. \tag{4.65}$$

Furthermore, it holds that for $m_j > l_i$, $m_j \mapsto \|\mathbf{E}\|m_j/(m_j - l_i)$ decreases monotonously as m_j increases.

For the second inequality, observe that for $m_j = \frac{1}{2}l_i$,

$$\frac{\|\mathbf{E}\|m_j}{|l_i - m_j|} = \frac{\frac{1}{2}\|\mathbf{E}\|l_i}{l_i - \frac{1}{2}l_j} = \|\mathbf{E}\|, \tag{4.66}$$

and if $m_j < l_i$, $m_j \mapsto \|\mathbf{E}\|m_j/(l_i - m_j)$ is decreasing monotonously as m_j decreases. ■

Based on the last lemma, we can bound the sum (4.60) as follows:

Lemma 4.67 *Define the set*

$$J(l_i) = \Big\{ j \in \{1, \ldots, r\} \;\Big|\; \frac{1}{2}l_i \leq m_j \leq 2l_i \Big\}. \tag{4.68}$$

Then, with $C(l_i) = |J(l_i)|$,

$$\sum_{j=1}^{r} \omega_{ij} m_j \leq 2l_i C(l_i) + 2r\|\mathbf{E}^r\|. \tag{4.69}$$

Proof It holds that

$$\sum_{j=1}^{r} \omega_{ij} m_j = \sum_{j \in J(l_i)} \omega_{ij} m_j + \sum_{j \notin J(l_i)} \omega_{ij} m_j. \tag{4.70}$$

For $j \in J(l_i)$, $\omega_{ij} m_j \leq m_j \leq 2l_i$, and for $j \notin J(l_i)$, $\omega_{ij} m_j \leq 2\|\mathbf{E}\|$ by the previous lemma. Therefore,

$$\sum_{j=1}^{r} \omega_{ij} m_j \leq \sum_{j \in J(l_i)} 2l_i + \sum_{j \notin J(l_i)} 2\|\mathbf{E}\| \leq 2l_i C(l_i) + (r - C(l_i))\|\mathbf{E}\|. \tag{4.71}$$

Since $C(l_i)$ will be rather small typically, we can simplify the bound by omitting the second occurrence of the $C(l_i)$ term. This completes the proof of the lemma. ■

Note that of the two terms in (4.69), only the first term $2l_i C(l_i)$ does not scale with $\|\mathbf{E}^r\|$. This term relates to the number of eigenvalues which cluster around l_i. Therefore, we see that the perturbation is basically confined to the cluster around l_i.

4.9 Truncating the Function

For degenerate kernels, functions f in the image of T_k always have a finite expansion in terms of the r eigenfunctions $\psi_1, \ldots, \psi_r$ of k. For general kernels, there is an infinite number of eigenfunctions, and the expansion is an infinite series. It is not feasible to treat all terms individually. As we have seen in the last chapter, for each eigenfunction, we have to consider the pseudo-inverse of the matrix $\mathbf{\Psi}$, where the columns of $\mathbf{\Psi}$ are the sample vectors of the first r eigenfunctions. But the more eigenfunctions are used to construct $\mathbf{\Psi}$, the more sample points are necessary such that $\mathbf{\Psi}^+$ is small. We will therefore use an approach similar to the treatment of kernel matrices of general kernel functions and truncate f, estimating the truncation error. We will assume that the kernel function is regular in the sense that it is uniformly bounded by K.

We consider functions which lie in the span of (ψ_i), the eigenfunctions of T_k. Since these are orthogonal, the norm of a function $g = \sum_{i=1}^{\infty} \beta_i \psi_i$ is

$$\|g\|^2 = \left\| \sum_{i=1}^{\infty} \beta_i \psi_i \right\|^2 = \sum_{i=1}^{\infty} \beta_i^2 \|\psi_i\|^2 = \sum_{i=1}^{\infty} \beta_i^2, \tag{4.72}$$

by Parseval's equation.

First of all, we prove that the truncation error cannot become larger than the norm of the original function.

Lemma 4.73 *Let $g = \sum_{i=1}^{\infty} \beta_i \psi_i$ with $(\beta_i) \in \ell^2$ and (ψ_i) being the eigenvectors of T_k which form an orthogonal set. Then,*

$$\|g - g^{[r]}\| \leq \|g\|, \tag{4.74}$$

where $g^{[r]} = \sum_{\ell=1}^{r} \beta_i \psi_i$.

Proof Note that

$$\|g - g^{[r]}\|^2 = \left\| \sum_{i=r+1}^{\infty} \beta_i \psi_i \right\|^2 = \sum_{i=r+1}^{\infty} \beta_i^2 \leq \sum_{i=1}^{\infty} \beta_i^2 = \|g\|^2, \tag{4.75}$$

since the terms β_i^2 are all positive. ■

We consider a function $f \in \operatorname{ran} T_k$ with $f = \sum_{i=1}^{\infty} \alpha_i \lambda_i \psi_i$. For bounded kernel functions, one can use the previous lemma to show that f and also $f - f^{[r]}$ are uniformly bounded.

Lemma 4.76 *Let k be a kernel function bounded by $K < \infty$, and $f = T_k g$. Then, for all $x \in \mathcal{X}$*

$$|f(x)| \leq K\|g\|, \qquad |f(x) - f^{[r]}(x)| \leq K\|g\|. \tag{4.77}$$

Proof Fix $x \in \mathcal{X}$. Let $k_x(y) = k(x, y)$. Then,

$$|f(x)| = |T_k g(x)| = \left| \int_{\mathcal{X}} k(x, y) g(y)\, \mu(dy) \right| = |\langle k_x, g \rangle_\mu| \leq \|k_x\| \|g\| \leq K\|g\|, \tag{4.78}$$

by the Cauchy-Schwarz inequality and since $\|k_x\| = (\int_{\mathcal{X}} k^2(x, y)\, \mu(dy))^{1/2} \leq K$.

Analogously, since $f^{[r]} = T_k g^{[r]}$,

$$|f(x) - f^{[r]}(x)| = |T_k(g(x) - g^{[r]}(x))| \leq K\|g - g^{[r]}\| \leq K\|g\|, \tag{4.79}$$

using Lemma 4.73. ■

By the strong law of large numbers,

$$\frac{1}{n}\|f(\mathbf{X}) - f^{[r]}(\mathbf{X})\|^2 = \frac{1}{n}\sum_{i=1}^{n}\left(f(X_i) - f^{[r]}(X_i)\right)^2 \to \mathsf{E}\left((f(X_i) - f^{[r]}(X_i))^2\right) = \|f - f^{[r]}\|^2 \tag{4.80}$$

almost surely. We want to derive finite sample size bounds for the convergence error. First we compute the relevant probabilistic properties.

Lemma 4.81 *Let $f \in \operatorname{ran} T_k$. Then, there exists an $F < \infty$ such that $|f(x)| \leq F$ for all $x \in \mathcal{X}$, and*

$$\mathsf{E}\left(\frac{1}{n}\sum_{i=1}^{n}\|f(\mathbf{X}) - f^{[r]}(\mathbf{X})\|^2\right) = \|f - f^{[r]}\|^2 = \sum_{j=r+1}^{\infty} \alpha_j^2 \lambda_j^2, \tag{4.82}$$

$$\mathsf{Var}_\mu\left((f - f^{[r]})^2\right) \leq F^2 \sum_{j=r+1}^{\infty} \alpha_j^2 \lambda_j^2. \tag{4.83}$$

Proof Existence of F follows from Lemma 4.76. The expectation of the finite sample size error has just been computed above. Then,

$$\|f - f^{[r]}\|^2 = \sum_{j=r+1}^{\infty} \alpha_j^2 \lambda_j^2 \tag{4.84}$$

follows from (4.72).

By the definition of the variance and the Hölder inequality,

$$\begin{aligned} \mathsf{Var}_\mu\left((f - f^{[r]})^2\right) &= \mathsf{E}\left((f - f^{[r]})^4\right) - \left(\mathsf{E}(f - f^{[r]})^2\right)^2 \leq \mathsf{E}\left((f - f^{[r]})^4\right) \\ &\leq \|(f - f^{[r]})^2\|_\infty \|(f - f^{[r]})^2\|_1 = F^2 \sum_{j=r+1}^{\infty} \alpha_j^2 \lambda_j^2, \end{aligned} \tag{4.85}$$

since $(f(x) - f^{[r]}(x))^2 \leq f^2(x) \leq F^2$ also by Lemma 4.76. ■

We see that the squared sum of $(\alpha_j \lambda_j)$ determines the size of the truncation error. For convenience, we define

$$T_r = \sqrt{\sum_{j=r+1}^{\infty} \alpha_j^2 \lambda_j^2}. \tag{4.86}$$

Based on the estimates from the last lemma, we can derive a large deviation bound for the truncation error.

Lemma 4.87 *Let $f \in \operatorname{ran} T_k$ and $F < \infty$ be such that $|f(x)| \leq F$ for all $x \in \mathcal{X}$, and T_r as defined above. Then, for $0 < \delta < 1$, with probability larger than $1 - \delta$*

$$\frac{1}{n}\|f(\mathbf{X}) - f^{[r]}(\mathbf{X})\|^2 < \|f - f^{[r]}\|^2 + F T_r \sqrt{\frac{1}{n\delta}}. \tag{4.88}$$

Proof By the Chebychev inequality (Theorem 2.38), for $\varepsilon > 0$,

$$\mathsf{P}\left\{\frac{1}{n}\|f(\mathbf{X}) - f^{[r]}(\mathbf{X})\|^2 - \|f - f^{[r]}\|^2 \geq \varepsilon\right\} \leq \frac{\mathsf{Var}_\mu\left((f - f^{[r]})^2\right)}{n\varepsilon^2} \leq \sqrt{\frac{F^2 T_r^2}{n\varepsilon^2}}. \tag{4.89}$$

Setting the right hand side equal to δ and solving for ε proves the lemma. ■

In principle, one can obtain a comparable bound based on the Bernstein inequality. However, similar to the situation encountered in Chapter 3, the random variable we are studying here has typically much smaller variance than the size of the range. The Bernstein inequality will give rise to an $O(n^{-1})$ term which scales with the range such that the resulting bound is less practical for small sample sizes. Therefore, we will use this bound based on the Chebychev inequality, which also meets our needs sufficiently well.

4.10 The Main Result

We have finally collected the necessary results to continue our analysis of the general case which has so far ended in the decomposition in Lemma 4.28. We repeat the final decomposition here and highlight the two parts of it:

$$\frac{1}{\sqrt{n}}|u_i^\top f(\mathbf{X})| \leq \sum_{\ell=1}^{r} |\alpha_\ell| \underbrace{\sum_{j=1}^{r} |u_i^\top v_j| \frac{1}{\sqrt{n}} |\lambda_\ell \psi_\ell(\mathbf{X})^\top v_j|}_{\text{(I) bound for truncated function } f^{[r]}} + \overbrace{\frac{1}{\sqrt{n}}\|f(\mathbf{X}) - f^{[r]}(\mathbf{X})\|}^{\text{(II) estimate of truncation error}}. \tag{4.90}$$

Recall that (l_i, u_i) are the eigenpairs of $\mathbf{K}$, and (m_j, v_j) are the corresponding pairs of $\mathbf{K}^{[r]}$. In this section, we write the eigenfunction sample matrix $\mathbf{\Psi}$ from Section 4.7 as $\mathbf{\Psi}^r$ to highlight the fact that it is built using only the first r eigenfunctions. Furthermore, for $f = \sum_{\ell=1}^{\infty} \alpha_\ell \lambda_\ell \psi_\ell$, denote the 1-norm of the sequence of coefficients α_ℓ truncated to the first r terms by

$$\|\alpha^{[r]}\|_1 = \sum_{\ell=1}^{r} |\alpha_\ell|. \tag{4.91}$$

Finally, the theorem will also refer to T_r, the asymptotic truncation error of f defined in (4.86), and $C(l_i)$, the number of eigenvalues of $\mathbf{K}^{[r]}$ lying between $2l_i$ and $\frac{1}{2}l_i$ defined in Lemma (4.67).

Theorem 4.92 *The scalar products between the sample vector of a function $f = \sum_{\ell=1}^{\infty} \alpha_\ell \lambda_\ell \psi_\ell$ with $(\alpha_\ell) \in \ell^2$, $|f(x)| \le F < \infty$, and the eigenvectors u_i of the kernel matrix can be bounded as follows*

$$\frac{1}{\sqrt{n}} |u_i^\top f(\mathbf{X})| < l_i C(r,n) + E(r,n) + T(r,n), \tag{4.93}$$

with probability larger than $1-\delta$, where

$$C(r,n) = 2\|\alpha^{[r]}\|_1 \|\mathbf{\Psi}^{r+}\| C(l_i), \tag{4.94}$$

$$E(r,n) = 2r\|\alpha^{[r]}\|_1 \|\mathbf{\Psi}^{r+}\| \|\mathbf{E}^r\|, \tag{4.95}$$

$$T(r,n) = T_r + \sqrt{FT_r} \sqrt[4]{\frac{1}{n\delta}}. \tag{4.96}$$

Proof First, we consider the part (I) (as in (4.90)). By Theorem 4.34,

$$\frac{1}{\sqrt{n}} |\lambda_\ell \psi_\ell(\mathbf{X})^\top v_j| \le m_j \|\mathbf{\Psi}^{r+}\|, \tag{4.97}$$

where $\mathbf{\Psi}^r$ is the same matrix as in (4.33), which is obtained from $\mathbf{K}^{[r]}$. (Here, we have silently assumed that $\mathbf{\Psi}^r$ is invertible, which is typically true as soon as $n > r$ is large enough.)

Furthermore, by Corollary 4.57,

$$|u_i^\top v_j| \le \frac{\|\mathbf{E}^r\|}{|l_i - m_j|} \wedge 1 = \omega_{ij}, \tag{4.98}$$

such that

$$\text{(I)} \le \|\mathbf{\Psi}^{r+}\| \sum_{j=1}^{r} \omega_{ij} m_j. \tag{4.99}$$

We have derived the following inequality in Lemma 4.67:

$$\sum_{j=1}^{r} \omega_{ij} m_j \le 2l_i C(l_i) + 2r\|\mathbf{E}^r\|. \tag{4.100}$$

Consequently,

$$\sum_{\ell=1}^{r} |\alpha_\ell| \sum_{j=1}^{r} |u_i^\top v_j| \frac{1}{\sqrt{n}} |\lambda_\ell \psi_\ell(\mathbf{X})^\top v_j| \le \sum_{\ell=1}^{r} |\alpha_\ell| \|\mathbf{\Psi}^{r+}\| \big(2l_i C(l_i) + 2r\|\mathbf{E}^r\|\big). \tag{4.101}$$

Next, we consider the part (II) which basically is the truncation error of the function f. By Lemma 4.87, with probability larger than $1-\delta$,

$$\frac{1}{n}\|f(\mathbf{X}) - f^{[r]}(\mathbf{X})\|^2 < \|f - f^{[r]}\|^2 + FT_r \sqrt{\frac{1}{n\delta}}. \tag{4.102}$$

Consequently,

$$\frac{1}{\sqrt{n}}\|f(\mathbf{X}) - f^{[r]}(\mathbf{X})\| < \sqrt{\|f - f^{[r]}\|^2 + FT_r \sqrt{\frac{2}{n}\log\frac{1}{\delta}}} \le \|f - f^{[r]}\| + \sqrt{FT_r} \sqrt[4]{\frac{1}{n\delta}}, \tag{4.103}$$

since $\sqrt{a^2+b^2} \le |a| + |b|$. Rearraging the terms proves the theorem. ■

4.11 Discussion

We end this chapter with a discussion of the results we have derived. The discussion will focus on three topics. First, we discuss the basic result on degenerate kernels and sample vectors of eigenfunctions, then, the main result for the general case. Finally, we undertake a comparison between the results presented here and the sampling theorem.

4.11.1 Degenerate Kernels and Eigenfunctions

The first result states that the properly scaled scalar product between the scaled eigenfunction sample vector $\lambda_\ell \psi_\ell(\mathbf{X})$ and an eigenvector u_i of the kernel matrix $\mathbf{K}_n$ scales as l_i with i. In other words, $\lambda_\ell \psi_\ell(\mathbf{X})$ is nearly orthogonal to the u_i for $i > \ell$. From the general convergence result, this situation is not clear, because, in principle, the eigenvector can be perturbed in any direction, also those spanned by eigenvectors of smaller eigenvalues. This result is already the main observation of this chapter.

In the asymptotic setting, the eigenfunction ψ_ℓ is orthogonal to all eigenfunctions ψ_m for $m \neq \ell$. Our result states that in the finite sample setting, this fact is still approximately true, because the sample vector is still almost orthogonal to eigenvectors of eigenvalues which are smaller than λ_ℓ. Thus, the location of $\lambda_\ell \psi_\ell$ with respect to the basis of eigenvectors of $\mathbf{K}$ is such that ψ_ℓ is essentially contained in the first ℓ eigenvectors of $\mathbf{K}$.

If one equates the complexity of an eigenvector to the size of the corresponding eigenvalue, which is plausible, because the inverse of the eigenvalue can be used to bound the regularity of the eigenfunction (see Section 4.5), the result can also be interpreted in the following manner: Since the sample vector of the eigenfunction is orthogonal to eigenvectors of smaller eigenvalues, the eigenfunction does not appear more complex on a finite sample than in the asymptotic case. Given a typical sample the sample vector of the eigenfunction will appear to be roughly as complex as the true eigenfunction. We will comment on this observation below when we discuss relations to the sampling theorem.

Note that this stability of regularity holds although the eigenvectors of smaller eigenvalues are in general more unstable than those of larger eigenvalues. Still, the perturbation is such that the sample vector will be almost orthogonal to these perturbations. Inverting the argument concerning the regularity, one can thus say that although the eigen*vectors* might not be stable their overall regularity will be constant and different perturbations of the same eigenvector will roughly have the same regularity.

4.11.2 The Relative-Absolute Envelope

In the transition from the degenerate to the general case, the purely relative envelope acquires an additional additive term, similar as in the discussion of the eigenvalues in Chapter 3. The resulting general relative-absolute envelope basically consists of two terms, namely $l_i C(r,n)$, which scales with l_i as in the degenerate case, and $E(r,n) + T(r,n)$ which does not vanish as $n \to \infty$.

Let us briefly consider the individual terms of $C(r,n) = 2\|\alpha^{[r]}\|_1 \|\mathbf{\Psi}^+\| \|C(l_i)$. The term $\|\alpha^{[r]}\|_1$ is linked to the complexity of f and will be discussed below. For the norm of the pseudo-inverse of $\mathbf{\Psi}$, we have seen in Lemma 4.44, that $\|\mathbf{\Psi}\| \to 1$, such that the term will become small as $n \to \infty$. Furthermore, $C(l_i)$ depends on the rate of decay of the eigenvalues. In the best case, for exponential decay $e^{-\beta i}$ with $\beta \geq \log 2$, $C(l_i) \to 1$, but $C(l_i)$ will be rather small also in general. All of these terms therefore being reasonably small, we conclude that the first term shows that the scalar products are of the order of $O(l_i)$ with a reasonable constant.

Since the additive term basically measures truncation errors, it holds that $E(r,n)+T(r,n) \to 0$ as $r \to \infty$. Therefore, for an appropriate choice of r, the additive term will become negligibly small and the whole bound forms an essentially relative envelope, showing that the scalar products decay quickly.

The complexity of f is basically measured in two ways, the 1-norm $\|\alpha^{[r]}\|_1$ of the first r coefficients of f, and the truncation error $f - f^{[r]}$. It is instructive to consider the complexity of

individual eigenfunctions ψ_ℓ in this setting. We assume that $r > \ell$. Then, $\psi_\ell - \psi_\ell^{[r]} = 0$. On the other hand,

$$\psi_\ell = \lambda_\ell \frac{1}{\lambda_\ell} \psi_\ell, \tag{4.104}$$

and we see that the (α_i) sequence corresponding to ψ_ℓ is

$$(\alpha_i) = \delta_{i\ell} \frac{1}{\lambda_\ell}. \tag{4.105}$$

Thus, eigenfunctions with small eigenvalues are more complex (in the sense that they lead to a larger bound), because

$$\|\psi_\ell^{[r]}\|_1 = \frac{1}{\lambda_\ell}. \tag{4.106}$$

Equivalent in complexity are $\lambda_\ell \psi_\ell$, which also coincides with our previous notion of complexity (Section 4.5), where complexity was measured in the norm of the pre-image under T_k, and $\lambda_\ell \psi_\ell = T_k \psi_\ell$, with $\|\psi_\ell\| = 1$.

4.11.3 Comparison to the Sampling Theorem

In this section, we will discuss the relation of the relative-absolute envelope to the sampling theorem (Shannon, 1949; Kotel'nikov, 1933). The sampling theorem states that a bandwidth limited function on $\mathbb{R}$ can be reconstructed perfectly from its values at k/W for $k \in \mathbb{Z}$, where W is the bandwidth. We discuss a considerably more simple case, that of periodic functions. In this case, a bandwidth limited function is constructed from only a finite number of basis functions. We argue that the relative-absolute envelope result can be interpreted as a generalization of this case to that of non-equidistant sample points and arbitrary orthogonal function sets.

Let $\mathcal{H}$ be a Hilbert space and $(\psi_\ell)_{\ell=1}^\infty$ an orthonormal system in this Hilbert space. An example might be $[0, 2\pi]$, and ψ_ℓ forming a sine basis. Now consider a function f which can be constructed using only finitely many basis functions

$$f(x) = \sum_{\ell=1}^{r} \alpha_\ell \psi_\ell(x). \tag{4.107}$$

In this situation, the complexity of f is finite in the sense that a finite number of points suffice to reconstruct f perfectly. In particular, any r points $x_1, \ldots, x_r$ will be sufficient as long as the $r \times r$ matrix

$$[\mathbf{\Psi}]_{i\ell} = \psi_\ell(x_i) \tag{4.108}$$

is invertible, and

$$\alpha_\ell = [\mathbf{\Psi}^{-1} f(\mathbf{x}^{(r)})]_\ell, \tag{4.109}$$

where $f(\mathbf{x}^{(r)}) = (f(x_1), \ldots, f(x_r))^\top$. Then,

$$f(x) = \sum_{\ell=1}^{r} [\mathbf{\Psi}^{-1} f(\mathbf{x}^{(r)})]_\ell \psi_\ell(x). \tag{4.110}$$

The situation is particularly simple if for every $n \in \mathbb{N}$, one can design n points $x_1, \ldots, x_n$, such that the sample vectors $\psi_\ell(\mathbf{x}^{(n)}) = (\psi_\ell(x_1), \ldots, \psi_\ell(x_n))^\top$ are orthogonal. Then,

$$\hat{\alpha}_\ell^{(n)} = \frac{\psi_\ell(\mathbf{x}^{(n)})^\top f(\mathbf{x}^{(n)})}{\psi_\ell(\mathbf{x}^{(n)})^\top \psi_\ell(\mathbf{x}^{(n)})}. \tag{4.111}$$

An example for such a setting is the following. Let $\mathcal{H} = L^2([0, 2\pi])$. On this space, we consider the orthogonal family of functions

$$\psi_\ell(x) = \frac{1}{\sqrt{\pi}} \sin(\ell x/2). \tag{4.112}$$

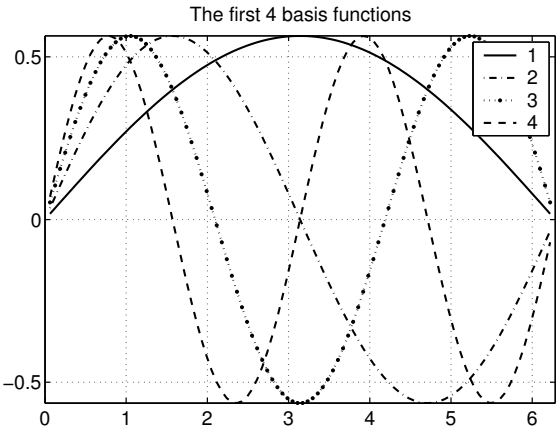


Figure 4.3: The first 4 basis functions $\psi_\ell(x) = \sin(\ell x/2)/\sqrt{\pi}$. The basis consists of sine waves with increasing frequency.

In Figure 4.3, the first four basis functions are plotted. Now, using equally spaced points for $1 \leq i \leq r$,

$$x_i = \frac{2\pi i}{2(r+1)}, \tag{4.113}$$

we obtain sample vectors

$$\psi_\ell(x_i) = \frac{1}{\sqrt{\pi}} \sin\left(\frac{\pi i \ell}{r+1}\right), \tag{4.114}$$

which are again orthogonal (although not normalized to unit length).

If such point sets can be constructed for arbitrary n, we readily obtain that

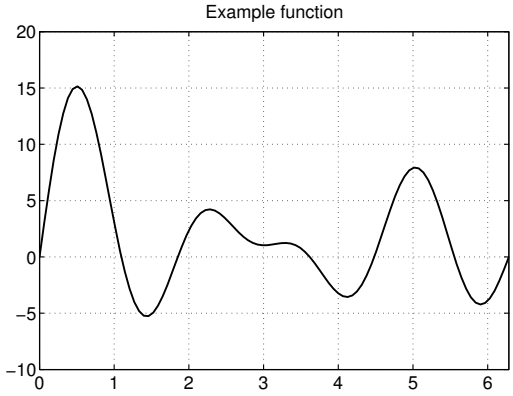
$$\hat{\alpha}_\ell^{(n)} = \begin{cases} \alpha_\ell & 1 \leq \ell \leq r, \\ 0 & \ell > r, \end{cases} \tag{4.115}$$

the latter because the sample vectors $\psi_\ell(\mathbf{x})$ are always orthogonal. Therefore, not are r points sufficient to recover f perfectly, but no further information can be gained using more sample points. In Figure 4.4, this is illustrated using a function which uses the first eight basis functions. The computed coefficients are constant after more than 8 sample points. In this sense, the described setting can be interpreted as a simplified version of the general sampling theorem (which treats non-periodic functions defined on $\mathbb{R}$, a situation which is considerably more involved). This similarity is also illustrated if one plots the contribution of $f(x_i)$ to the reconstruction (see Figure 4.5) which are very similar to the sinc functions occurring in the sampling theorem.

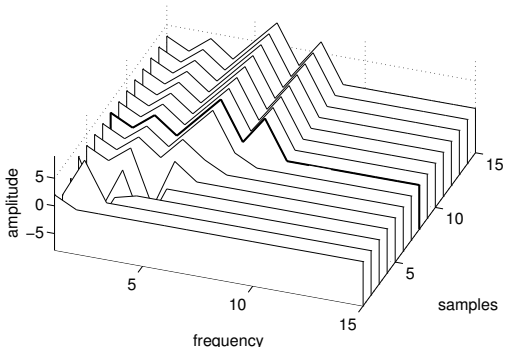
A bit more abstractly, we have the following situation: we have a basis of functions ψ_ℓ which have discrete counterparts given by the orthogonal basis $u_1, \ldots, u_n$. In this setting, the sampling theorem can be rephrased as follows: The important property of this correspondence is that if a function f uses only the first r functions ψ_ℓ, then the same holds with respect to the basis vectors u_ℓ:

	continuous	discrete
orthogonal basis	(ψ_ℓ)	u_ℓ
band limited function	$f = \sum_\ell \alpha_\ell \psi_\ell$	$f(\mathbf{x}) = \sum_\ell \hat{\alpha}_\ell u_\ell$
coefficients	$\alpha_\ell = 0,\ l > r$	$\hat{\alpha}_\ell = 0,\ l > r$

For the sine basis on $[0, 2\pi]$, it was easy to achieve this configuration, because the orthogonal basis could be obtained by sampling ψ_ℓ at equidistant points.



(a) An example function with coefficients 3, 1, 4, 1, 5, 9, 2, 7. Only the first eight basis functions are used resulting in a bandwidth limited function.



(b) The computed coefficients using 1 to 15 sample points.

Figure 4.4: Computing the coefficients of a function which uses the first eight basis functions for increasing numbers of equidistant sample points. After eight sample points, the coefficients stay the same (coefficients for eight sample points are highlighted).

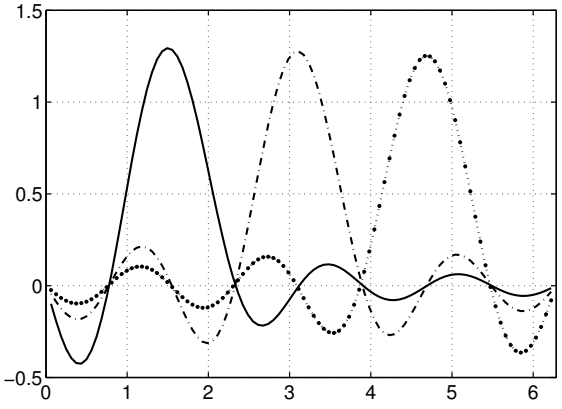


Figure 4.5: The contribution of the value of f at three sample points to the reconstructed function using the first eight coefficients. The resulting functions are very similar to the sinc functions occurring in the sampling theorem.

Let us now relate the setting discussed in this chapter to this framework. Instead of the sine basis on $[0, 2\pi]$, we have an arbitrary orthogonal family of functions (ψ_ℓ) on $\mathcal{H}_\mu(\mathcal{X})$. Instead of equally spaced points, we have an i.i.d. sample from some probability distribution μ. The basis functions correspond to the eigenvectors of the kernel matrix in the discrete case. These give approximate coefficients $\hat{\alpha}_\ell$. The question now is if a function f which uses only finitely many of the basis function has also only finitely many coefficients $\hat{\alpha}_\ell$ which are non-zero.

Theorem 4.92 shows that this is not true in the exact sense, but that $\hat{\alpha}_\ell$ decays quickly for as $O(\lambda_\ell/\lambda_r)$ for $\ell > r$, plus a small absolute term. Therefore, although it does not hold that $\hat{\alpha}_\ell = 0$ for $\ell > r$, nevertheless, the coefficients will be quite small. Therefore, this result has a similar interpretation as in the case of bandwidth limited functions: the number of non-zero coefficients for finitely many sample points is approximately the same as in the continuous case. With respect to the number of necessary sample points to reconstruct f perfectly, we obtain a different answer, though. The quality of the reconstruction is mainly governed by how good the u_ℓ approximate the true basis functions. These approximations become typically good only after a large number of data points have been sampled. However, the number of eigenvectors which has to be considered is given by r, such that functions which use many basis functions need considerably more data points to be reconstructed well.

In summary, in the sense explained so far, the relative-absolute envelope result can be interpreted as an analogue to the observation stemming from the sampling theorem that the complexity of the sample vector of a function is roughly the same as in the asymptotic case. This does not directly translate into a number of points necessary to reconstruct a function well, but there exists a connection to the number of eigenvectors of the kernel matrix which need to be good approximations of the true eigenfunctions. This connection again implies a relation between the number of sample points necessary to reconstruct a function with a given complexity. An analysis of the connection would be an interest direction for future research.

4.12 Conclusion

In this section, we have investigated the spectral projections of the kernel matrix. By the result by Koltchinskii (1998), these converge to the true spectral projections of the integral operator T_k. A slightly weaker question is that of scalar products between eigenvectors and sample vectors of fixed functions. First of all, the result on spectral projections implies the convergence for scalar products. However, simulations show that one could expect a different behavior, namely that the scalar products with eigenvectors to small eigenvalues are a priori bounded by a constant times the size of the eigenvalue. We have derived a relative-absolute envelope which proves that this is actually true. This concludes our investigations concerning the spectral structure of the kernel matrix.

In Chapter 6, we will see that these results here have interesting consequences in connection with the regression problem in supervised learning.

Part II

Applications to Kernel Methods

Chapter 5

Principal Component Analysis

Abstract

Principal Component Analysis (PCA) and its kernel based extension, kernel PCA, are unsupervised methods which analyze the structure of a vectorial sample, identifying uncorrelated directions of maximal variance. Since the PCA is based on the spectral decomposition of the covariance matrix, the results on the convergence of eigenvalues can readily be applied to a theoretical analysis of both PCA and kernel PCA. We show that the estimated principal values (the variances along the principal components) approximate the true principal values with high precision if the sequence of principal values decays quickly. This allows us to make predictions about the structure of a finite sample in feature space to the effect that a typical sample in a (possibly) infinite-dimensional feature space is contained in a low-dimensional subspace whose dimension does not depend on the sample size.

5.1 Introduction

Principal component analysis (PCA) is a standard method for analyzing vectorial data (Jolliffe, 2002). The goal is to find directions along which the variance of the data is maximal and the random vector projected to these directions are uncorrelated. These directions are called *principal directions* and the variances along these directions are called *principal values*. Asymptotically, these directions are given by the eigenvectors of the covariance matrix of the random vector. For a given finite sample, these directions are approximated by computing the eigenvectors of the sample covariance matrix.

For a normally distributed random vector, asymptotic results are well-known (for example, (Anderson, 1963)). In this case, one can even compute the distribution of the covariance matrix in closed form, known as the Wishart-distribution (Anderson, 2003). Asymptotic results exist also for general distributions, see for example (Dauxois et al., 1982). The result is then stated as a combination of a strong law of large number and a central limit theorem. In summary, for large sample sizes, we can expect that the estimated principal values converge to the true principal values.

Now since computationally, the principal component analysis amounts to computing the spectral decomposition of symmetric matrices, the results on eigenvalues from Chapter 3 are readily applicable. This means that we directly obtain convergence results for PCA. In the case of vectorial data in a finite-dimensional vector space, the error terms are even purely multiplicative.[1]

Many kernel methods arise as classical linear methods which act on a (possible infinite-dimensional) *feature space* into which the data has been mapped via a non-linear *feature map*. Performing PCA in feature spaces leads to the *kernel PCA* algorithm (Schölkopf et al., 1998).

[1]To the knowledge of the author, this result is new. In spite of the suggestive title, the paper by Chatterjee et al. (1998) discusses relative convergence properties not with respect to the approximation error, but relative in comparison with several algorithms.

The principal values of the data in feature space are given by the eigenvalues of the kernel matrix which again gives a direct connection to the results from Chapter 3.

The asymptotic behavior of kernel PCA has been a focus of recent research, which explores both how and if kernel PCA converges. In Shawe-Taylor and Williams (2003), the reconstruction error is considered which is the error induced by projecting the data to the space spanned by the first r principal directions. In Bengio et al. (2004), it is studied in how far the Nyström extrapolated principal directions reconstruct the kernel matrix. We present bounds based on the results from Chapter 3 which lead to tight bounds on the reconstruction error. These bounds have interesting implications for the structure of a finite sample showing that it is contained in a low-dimensional subspace in feature space up to a small error.

This chapter is structured as follows. Section 5.2 reviews the main results in a less technical manner. Section 5.3 contains the result for principal component analysis. The results for kernel PCA are derived in Section 5.4. Section 5.5 discusses implication for the effective dimension of data in feature space and the projection error in kernel PCA. Section 5.6 concludes this chapter.

5.2 Summary of Main Results

We present convergence results for PCA and kernel PCA, and discuss implications for the structure of a finite sample in feature space. Existing results are either central limit theorems or fail to reflect the fact that smaller principal values converge much faster. In contrast, the results developed in this chapter are non-asymptotic confidence bounds which result in much tighter error estimates if the principal values decay quickly. In machine learning, this is usually the case because the vectors encode measurements of real world objects which are often correlated.

For (linear) PCA in a finite-dimensional vector space we show that the principal values converge with relative error bounds.

Theorem 5.1 (Linear Finite-Dimensional PCA)
(Theorem 5.13 in the main text) The estimated principal values l_i converge to the true principal values λ_i with relative bounds

$$|l_i - \lambda_i| \leq \lambda_i \|\mathbf{R}_n - \mathbf{I}_d\|, \tag{5.2}$$

where $\mathbf{R}_n$ is the sample covariance matrix of $X_1, \ldots, X_n$ with $\mathbf{E}(X_i) = 0$ after a whitening step and a transformation such that the principal components lie along the coordinate axes.

For kernel PCA, we show that principal values converge with a relative-absolute error bound where the absolute error depends on the sum of all but the first few largest eigenvalues and is usually very small.

Theorem 5.3 (Kernel PCA in Infinite-Dimensional Feature Space)
(Theorem 5.39 in the main text) The estimated principal values l_i converge to the true principal values λ_i with a relative-absolute bound

$$|l_i - \lambda_i| \leq \lambda_i \|\mathbf{C}_n^r\| + \|\mathbf{E}_n^r\|, \tag{5.4}$$

where

$$\mathbf{C}_n^r = \mathbf{\Psi}_n^{r\top} \mathbf{\Psi}_n^r - \mathbf{I}_r, \qquad \mathbf{E}_n^r = \mathbf{K}_n - \mathbf{K}_n^{[r]}, \tag{5.5}$$

and the columns of $\mathbf{\Psi}_n^r$ are the sample vectors of the first r eigenvectors of the kernel matrix and $\mathbf{E}_n^r$ is the error matrix resulting from replacing the kernel function by its r-degenerate approximation.

The error matrices occurring in this theorem were introduced in Chapter 3 where the supporting theoretical results were derived. These must not be confused with dimension reduction errors occurring in kernel PCA. The r-degenerate approximation is performed with respect to the kernel function by replacing its infinite expansion in Mercer's formula by a finite sum.

These bounds contain a parameter r which lies between 1 and n and which roughly controls the number of eigenvalues for which the relative error is computed. As discussed in Chapter 3, there exists a trade-off between $\mathbf{C}_n^r$ and $\mathbf{E}_n^r$, such that there is no easy optimal choice. One usually considers r fixed such that $\|\mathbf{E}_n^r\|$ is small, around the precision of the underlying finite precision arithmetics involved.

The terms $\|\mathbf{C}_n^r\|$ and $\|\mathbf{E}_n^r\|$ have been extensively studied for two cases (kernels with bounded eigenfunctions and bounded kernel functions) in Sections 3.9 and 3.10. Asymptotically, $\|\mathbf{C}_n^r\| \to 0$ and $\|\mathbf{E}_n^r\|$ is upper bounded by $\sum_{i=r+1}^{\infty} \lambda_i^2$ (see Theorem 3.131).

This theorem improves upon results from Shawe-Taylor and Williams (2003) and Zwald et al. (2004) which only treat leading sums of eigenvalues and sums of all but the first few eigenvalues. Moreover, those bounds are independent of the number of eigenvalues or the magnitude of the true eigenvalues under consideration, overestimating the error if the eigenvalues are very small. The present theorem also addresses the finite-sample size case in contrast to the central limit theorems of Dauxois et al. (1982) and Koltchinskii and Giné (2000).

The result for kernel PCA has interesting implications for the structure of a finite-sample in feature space. A smooth kernel results in a feature map whose asymptotic principal values decay quickly. This means that there are only a few directions along which the variance of the data in feature space is large. Now in principle, it is conceivable that a finite sample effectively covers a much larger space than in the asymptotic case because the principal values converge only slowly due to a large fourth moment of the data. Since a sample of size n spans an n-dimensional subspace in feature space, in the worst case, the effective dimension of the data scales with the sample size, rendering the learning problem hard. However, Theorem 5.3 implies that the estimated principal values approximate the true ones with high precision such that they also decay quickly. Consequently, a finite sample is contained in a low-dimensional subspace. This can be expressed with respect to approximation bounds on the projection error of kernel PCA.

Theorem 5.6 (Projection error for kernel PCA)
(Theorem 5.54 in the main text) The squared projection error P_d^2 for the first d dimensions is bounded by

$$P_d^2 \leq (1 + \|\mathbf{C}_n^r\|)\Pi_d^2 + n\|\mathbf{E}_n^r\|, \tag{5.7}$$

where $\mathbf{C}_n^r$ and $\mathbf{E}_n^r$ are the same error matrices as in the Theorem 5.3.

Here, Π_d^2 is the asymptotic reconstruction error which is given by $\Pi_d^2 = \sum_{i=d+1}^{\infty} \lambda_i$. For fixed r, the absolute term scales as n, but note that $\|\mathbf{E}_n^r\|$ will typically be very small. As $n \to \infty$, r can be chosen accordingly, such that the bound converges to Π_d^2. Thus, this theorem amounts to a relative-absolute bound on the projection error which also results in a much tighter bound for large d compared to existing approaches.

5.3 Principal Component Analysis

We briefly review principal component analysis. Let X be a random variable in $\mathbb{R}^d$ with distribution μ. The *covariance matrix* $\mathbf{Cov}(X)$ is the $d \times d$ matrix with entries

$$[\mathbf{Cov}(X)]_{ij} = \mathbf{E}\big(([X]_i - \mathbf{E}[X]_i)([X]_j - \mathbf{E}[X]_j)\big). \tag{5.8}$$

The first principal component is given by the unit length vector $v \in \mathbb{R}^d$ which maximizes

$$\mathbf{Var}(X^\top v) = v^\top \mathbf{Cov}(X) v. \tag{5.9}$$

By the variational characterization of eigenvalues (see Theorem 3.35), it follows that the solution v_1 is given by an eigenvector to the largest eigenvalue of $\mathbf{Cov}(X)$. The second principal component is the vector which maximizes $v^\top \mathbf{Cov}(X) v$ with the constraint that $v_1 \perp v$, because $v^\top X$ should be uncorrelated from $v_1^\top X$. The solution is given by an eigenvector to the second largest eigenvalue of $\mathbf{Cov}(X)$, and so on.

Now given only a finite sample $X_1, \dots, X_n \in \mathbb{R}^d$, the (population) principal components are estimated using the sample covariance matrix

$$[\mathbf{C}_n]_{ij} = \frac{1}{n-1} \sum_{\ell=1}^{n} ([X_\ell]_i - [\overline{X}]_i)([X_\ell]_j - [\overline{X}]_j), \tag{5.10}$$

with $\overline{X} = \frac{1}{n} \sum_{\ell=1}^{n} X_\ell$. This matrix is called the *sample covariance matrix* (see, for example, Pestman (1998)). The sample covariance matrix can be thought of as an approximation of $\mathbf{Cov}(X)$ obtained by replacing the expectation by the empirical averages. The estimated principal components $u_1, \dots u_d$ are the eigenvectors of $\mathbf{C}_n$, and the estimated principal values $l_1, \dots, l_d$ are the eigenvalues of $\mathbf{C}_n$.

For simplicity, we will assume that $\mathbf{E}(X) = 0$. In this case, one can also consider the already centered sample covariance matrix

$$[\mathbf{C}_n]_{ij} = \frac{1}{n} \sum_{\ell=1}^{n} [X_\ell]_i [X_\ell]_j. \tag{5.11}$$

Since we are in a finite-dimensional setting, it is rather easy to prove that the principal values converge: Since $\mathbf{C}_n$ has only finitely many entries and by the strong law of large numbers $[\mathbf{C}_n]_{ij} \to [\mathbf{Cov}(X)]_{ij}$, $\|[\mathbf{C}_n]_{ij} - \mathbf{Cov}(X)_{ij}\| \to 0$, and consequently, the eigenvalues of $\mathbf{C}_n$ converge to those of $\mathbf{Cov}(X)$ by Weyl's theorem (Theorem 3.49),

$$|l_i - \lambda_i| \leq \|[\mathbf{C}_n]_{ij} - \mathbf{Cov}(X)_{ij}\|. \tag{5.12}$$

This bound is again absolute in the sense that the same bound is applied to all principal values, leading to a rather pessimistic error estimate for the smaller principal values.

We can obtain a relative perturbation bound using the same techniques as in Chapter 3. Here and in the following we will always assume that all principal values are non-zero. If this is not the case, one considers the subspace spanned by the principal directions corresponding to non-zero principal values, because samples lie in this subspace almost surely.

Theorem 5.13 *Let μ be a probability distribution on $\mathbb{R}^d$ with mean zero, non-zero principal values $\lambda_1, \dots, \lambda_d$, and principal directions $\psi_1, \dots, \psi_d$. Let $l_1, \dots, l_d$ be the estimated principal values and $u_1, \dots, u_d$ the estimated principal directions based on an i.i.d. sample $X_1, \dots, X_n$ from μ. Then,*

$$|l_i - \lambda_i| \leq \lambda_i \|\mathbf{R}_n - \mathbf{I}_d\|, \tag{5.14}$$

where $\mathbf{R}_n$ is the $d \times d$ matrix with entries

$$[\mathbf{R}_n]_{ij} = \frac{1}{\sqrt{\lambda_i \lambda_j}} \psi_i^\top \mathbf{C}_n \psi_j. \tag{5.15}$$

It holds that $\|\mathbf{R}_n - \mathbf{I}_d\| \to 0$ almost surely.

Proof Let $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \dots, \lambda_d)$ and $\mathbf{\Psi}$ be the $d \times d$ matrix whose columns are $\psi_1, \dots, \psi_d$. Then, $\mathbf{\Psi}^\top \mathbf{\Psi} = \mathbf{\Psi} \mathbf{\Psi}^\top = \mathbf{I}_d$, because the (ψ_i) are orthonormal (since they are the eigenvectors of the symmetric matrix $\mathbf{Cov}(X)$). Let $\mathbf{X}$ be the $d \times n$ matrix whose columns are the samples $X_1, \dots, X_n$. Using $\mathbf{X}$, the sample covariance from (5.11) can be written as

$$\mathbf{C}_n = \frac{1}{n} \mathbf{X} \mathbf{X}^\top. \tag{5.16}$$

Now $\mathbf{X}\mathbf{X}^\top$ has the same non-zero eigenvalues as $\mathbf{X}^\top \mathbf{X}$, and we can re-write $\mathbf{X}^\top \mathbf{X}/n$ as

$$\frac{1}{n} \mathbf{X}^\top \mathbf{X} = \frac{1}{n} \mathbf{X}^\top \mathbf{\Psi} \mathbf{\Psi}^\top \mathbf{X} = \left(\frac{1}{\sqrt{n}} \mathbf{X}^\top \mathbf{\Psi} \mathbf{\Lambda}^{-\frac{1}{2}} \right) \mathbf{\Lambda} \left(\frac{1}{\sqrt{n}} \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{\Psi}^\top \mathbf{X} \right). \tag{5.17}$$

Thus, $\mathbf{X}^\top\mathbf{X}/n$ can be considered a multiplicative perturbation of $\mathbf{\Lambda}$ to $\mathbf{S}\mathbf{\Lambda}\mathbf{S}^\top$ with

$$\mathbf{S} = \frac{1}{\sqrt{n}}\mathbf{X}^\top\mathbf{\Psi}\mathbf{\Lambda}^{-\frac{1}{2}}. \tag{5.18}$$

Since $\mathbf{\Lambda}$ has eigenvalues $\lambda_1, \ldots, \lambda_d$, it follows from Ostrowski's theorem (Theorem 3.52) that

$$|l_i - \lambda_i| \leq \lambda_i \|\mathbf{S}^\top\mathbf{S} - \mathbf{I}\|. \tag{5.19}$$

Let us compute the entries of $\mathbf{S}^\top\mathbf{S}$. First note that

$$[\mathbf{S}]_{ij} = \frac{1}{\sqrt{n\lambda_j}} X_i^\top \psi_j, \tag{5.20}$$

and consequently

$$[\mathbf{S}^\top\mathbf{S}]_{ij} = \sum_{\ell=1}^{n} \frac{1}{n\sqrt{\lambda_i\lambda_j}} X_\ell^\top\psi_i X_\ell^\top\psi_j = \frac{1}{n\sqrt{\lambda_i\lambda_j}} \sum_{\ell=1}^{n} \psi_i^\top X_\ell X_\ell^\top \psi_j = \frac{1}{n\sqrt{\lambda_i\lambda_j}} \psi_i^\top \mathbf{X}\mathbf{X}^\top\psi_j. \tag{5.21}$$

And re-substituting $\mathbf{C}_n = \mathbf{X}\mathbf{X}^\top/n$ yields (5.15).

With respect to convergence, note that

$$\frac{1}{\sqrt{\lambda_i\lambda_j}}\psi_i^\top\mathbf{C}_n\psi_j \to \frac{1}{\sqrt{\lambda_i\lambda_j}}\psi_i^\top \mathbf{Cov}(X)\psi_j \qquad \text{almost surely.} \tag{5.22}$$

Since ψ_i is an eigenvector of $\mathbf{Cov}(X)$,

$$\frac{1}{\sqrt{\lambda_i\lambda_j}}\psi_i^\top \mathbf{Cov}(X)\psi_j = \frac{\lambda_i}{\sqrt{\lambda_i\lambda_j}}\psi_i^\top\psi_j = \delta_{ij}, \tag{5.23}$$

and therefore $\mathbf{R}_n \to \mathbf{I}_d$. ■

Let us take a look at the error matrix $\mathbf{R}_n$. This matrix is really the sample covariance matrix of the random variable X after it has been transformed into the basis of principal components and scaled along the principal components such that the variance becomes 1. To see this, note first that $\psi_i^\top X_\ell$ occurring in formula (5.21) is X projected onto the ith principal direction. The variance along this direction is $\mathbf{Var}(\psi_i^\top X_\ell) = \lambda_i$. By dividing by $\sqrt{\lambda_i}$, the variance becomes 1. Using these projected and scaled versions of X, one obtains a random variable which has the same principal directions as X but principal values equal to 1 by

$$\sum_{i=1}^{d} \psi_i \frac{1}{\sqrt{\lambda_i}}\psi_i^\top X_\ell. \tag{5.24}$$

On the other hand, one could also replace the first term ψ_i in the sum by the standard basis of $\mathbb{R}^n$ and obtains the random variable

$$Z_\ell = \left(\frac{1}{\sqrt{\lambda_1}}\psi_1^\top X_\ell, \ldots, \frac{1}{\sqrt{\lambda_d}}\psi_d^\top X_\ell\right), \tag{5.25}$$

which is already rotated such that the principal directions are given by the coordinate axes. The sample covariance matrix of Z_ℓ is

$$\frac{1}{n\sqrt{\lambda_i\lambda_j}} \sum_{\ell=1}^{n} \psi_i^\top X_\ell \psi_j^\top X_\ell \tag{5.26}$$

which is just the same term as in (5.21). Therefore, the convergence depends on how fast the sample covariance matrix of the projected and scaled X converge to $\mathbf{I}_d$.

Contrast this result with the naive relative bound obtained by dividing by λ_i:

$$|l_i - \lambda_i| \leq \lambda_i \frac{\|\mathbf{C}_n - \mathbf{Cov}(X)\|}{\lambda_i}. \tag{5.27}$$

Here, the whole error term is scaled by λ_i, whereas in Theorem 5.13, the principal directions are scaled individually, such that the resulting error term really measures the error in each direction at the given relative scale.

5.4 The Feature Space and Kernel PCA

In the present section, we discuss bounds on the estimation error of principal values in the context of kernel PCA. We start by introducing the notion of feature space.

The notion of *feature space* is linked indivisibly to that of kernel methods. One can either characterize kernel methods as methods which construct a fit from a number of kernels centered around each data point, or, for certain kernel functions, one can alternatively interpret kernel methods as ordinary linear methods which act in a feature space. This approach is followed in a number of expositions to kernel methods, including (Schölkopf, 1997; Vapnik, 1998; Schölkopf and Smola, 2002; Herbrich, 2002).

A *feature space* is just another linear space $\mathcal{F}$, possibly of infinite dimension into which the original space is mapped in a non-linear fashion by some function Φ, which is called the *feature map*. A typical example would be $\Phi\colon \mathbb{R}^d \to \ell^2$.

The idea behind such a construction is that one can use some traditional linear method like linear regression on the transformed data, thereby increasing the expressive power of the method in a controlled way.

Of course, explicitly mapping vectors to $\mathcal{F}$ is computationally infeasible if the dimension of $\mathcal{F}$ is large or even infinite. Therefore, an important requirement of the pair $(\mathcal{F}, \Phi)$ defining the feature space is that the necessary vector space operations can be computed efficiently in feature space. For many applications, it suffices to be able to compute scalar products efficiently. In other words, there exists some efficiently computable function k such that for all $x, y \in \mathbb{R}^d$,

$$\langle \Phi(x), \Phi(y) \rangle_{\mathcal{F}} = k(x, y). \tag{5.28}$$

The usual way to define a feature space is thus by specifying such a k which computes a scalar product in some feature space. An important class of functions implicitly define a feature space are the Mercer kernels which are studied throughout this thesis. If k is a Mercer kernel with eigenvalues λ_i and eigenfunctions ψ_i, then

$$x \mapsto \Phi(x) = \left(\sqrt{\lambda_i}\psi_i(x)\right)_{i \in \mathbb{N}} \tag{5.29}$$

defines a feature map into ℓ^2. Scalar products are then given by:

$$\langle \Phi(x), \Phi(y) \rangle_{\ell^2} = \sum_{i=1}^{\infty} \lambda_i \psi_i(x) \psi_i(y) = k(x, y). \tag{5.30}$$

Now assume that one uses a Mercer kernel k to construct some fit

$$\hat{f}(x) = \sum_{i=1}^{n} k(x, X_i)\alpha_i. \tag{5.31}$$

Let us translate this in terms of a feature space:

$$\hat{f}(x) = \sum_{i=1}^{n} \langle \Phi(x), \Phi(X_i) \rangle \alpha_i = \left\langle \Phi(x), \sum_{i=1}^{n} \alpha_i \Phi(X_i) \right\rangle =: \langle \Phi(x), w \rangle. \tag{5.32}$$

We see that this fit corresponds to a linear fit in feature space. Therefore, kernel methods using a Mercer kernel correspond to linear methods in feature space. The process of turning a linear method into a method working in feature space is called *kernelization*. Note, however, that not all interesting kernel functions are Mercer kernels.

The extension of PCA to feature spaces is called kernel PCA (Schölkopf et al., 1998). The idea is to perform a PCA on $\Phi(X_1), \ldots, \Phi(X_n)$. Since PCA amounts to maximizing certain scalar products $v \mapsto \langle v, X \rangle$, a kernelization should be straightforward. However, we cannot simply translate the approach of computing the eigenvectors of $\mathbf{C}$, because $\mathbf{C}$ has the dimension of the space the data lives in, and will therefore be a potentially infinite-dimensional matrix.

The follow trick is used. Let $\mathbf{X}$ be the $d \times n$ matrix whose ith column is $X_i/\sqrt{n}$. Then, $\mathbf{C} = \mathbf{X}\mathbf{X}^\top$. On the other hand, the matrix $\mathbf{K} = \mathbf{X}^\top\mathbf{X}$ has size n, the same non-zero eigenvalues as $\mathbf{C}$, and the entries are given by the scalar-products between the X_i:

$$[\mathbf{K}]_{ij} = \frac{1}{n} X_i^\top X_j. \tag{5.33}$$

This matrix is finite-dimensional even if the X_i lie in an infinite-dimensional space. Now if v is an eigenvector of $\mathbf{K}$ to eigenvalue λ, then,

$$\mathbf{C}\mathbf{X}v = \mathbf{X}\mathbf{X}^\top\mathbf{X}v = \mathbf{X}\mathbf{K}v = \lambda\mathbf{X}v, \tag{5.34}$$

thus $\mathbf{X}v$ is an eigenvector of $\mathbf{C}$ to the same eigenvalue. We can therefore compute the eigenvectors of $\mathbf{C}$ by computing those of $\mathbf{K}$ and multiplying by $\mathbf{X}$.

Now consider $\Phi(X_1), \ldots, \Phi(X_n)$. Then, by (5.33),

$$[\mathbf{K}]_{ij} = \langle \Phi(X_i), \Phi(X_j) \rangle_{\mathcal{F}} = k(X_i, X_j). \tag{5.35}$$

Assume that $v \in \mathbb{R}^n$ is an eigenvector of $\mathbf{K}$. What is the feature space equivalent of $\mathbf{X}v$? For the finite-dimensional case,

$$\mathbf{X}v = \sum_{i=1}^n X_i [v]_i, \qquad \text{and analogously} \qquad \Phi(\mathbf{X})v := \sum_{i=1}^n \Phi(X_i)[v]_i. \tag{5.36}$$

Thus, $\Phi(\mathbf{X})v$ is an eigenvector of the covariance matrix in feature space. We want to normalize this eigenvector such that its length is 1:

$$1 \stackrel{!}{=} \|\Phi(\mathbf{X})v\|^2 = \langle \Phi(\mathbf{X})v, \Phi(\mathbf{X})v \rangle = \sum_{i,j} v_i k(X_i, X_j) v_j = v^\top \mathbf{K} v = \lambda \|v\|^2. \tag{5.37}$$

Therefore, if $\|v\| = 1$, as usually computed by standard methods for symmetric eigenvectors problems, then the eigenvectors in feature space will be $\Phi(\mathbf{X})v/\lambda$.

Let us compute the projection of a vector $y \in \mathbb{R}^d$ on its ith principal direction (again assuming that $\lambda_i \neq 0$):

$$\begin{aligned} \langle \Phi(y), \Phi(\mathbf{X})v_i/\lambda_i \rangle &= \frac{1}{\lambda_i} \left\langle \Phi(y), \sum_{j=1}^n \Phi(X_j)[v_i]_j \right\rangle = \frac{1}{\lambda_i} \sum_{j=1}^n \langle \Phi(y), \Phi(X_j) \rangle [v_i]_j \\ &= \frac{1}{\lambda_i} \sum_{j=1}^n k(y, X_j)[v_i]_j. \end{aligned} \tag{5.38}$$

We see that these projections can again be conveniently computed using the kernel function[2].

We have seen why results on the eigenvalue of the kernel matrix directly translate to results for kernel PCA. This connection has of course been exploited before, and in principle all results reviewed in Section 3.4 generate a corresponding result for kernel PCA, for which our comments from Section 3.4 apply accordingly. In terms of non-asymptotic error bounds, so far only bounds for sums of eigenvalues were known. These bounds allow us to conveniently analyze the reconstruction

[2] As has been re-discovered recently by Bengio et al. (2004), (5.38) computes approximations to the eigenfunctions of T_k. More precisely, the last expression in (5.38) is an eigenfunction of the operator

$$f \mapsto \frac{1}{n} \sum_{i=1}^n k(x, X_i) f(X_i),$$

also known as the *Nyström approximation* of T_k using Monte Carlo integration to approximate the integral. These types of approximation are commonplace in the numerical approximation of integral equations and are known at least since Nyströms initial paper Nyström (1930). For an overview, see (Anselone, 1971) or (Engl, 1997) (in German), which also contains alternative derivations of the Propositions 1 and 2 from Bengio et al. (2004).

error which is the error induced by projecting the sample to the first r principal directions. In Shawe-Taylor and Williams (2003) such a result is mentioned with the proof delegated to Shawe-Taylor et al. (2004). However, the bound does not scale well with r, namely, the (absolute!) confidence term scales as $O(\sqrt{r})$ and does not depend on the sum of the eigenvalues smaller than λ_r.

Thus, we obtain the final result for kernel PCA by rephrasing the respective result (Theorem 3.71) from Chapter 3:

Theorem 5.39 (Relative-Absolute Bound for the Estimated Principal Values in Kernel PCA) *Let $(l_i)_{i=1}^n$ be the estimated principal values for kernel PCA using a Mercer kernel k with eigenvalues $(\lambda_i)_{i=1}^\infty$ and eigenfunctions $(\psi_i)_{i=1}^\infty$. Then, the principal values l_i converge to the eigenvalues λ_i with the following relative-absolute bound (for $1 \leq r \leq n$),*

$$|l_i - \lambda_i| \leq \lambda_i \|\mathbf{C}_n^r\| + \|\mathbf{E}_n^r\|, \tag{5.40}$$

where

$$\mathbf{C}_n^r = {\mathbf{\Psi}_n^r}^\top \mathbf{\Psi}_n^r - \mathbf{I}_r, \qquad \mathbf{E}_n^r = \mathbf{K}_n - \mathbf{K}_n^{[r]}. \tag{5.41}$$

The error terms are the same as in Chapter 3: $\mathbf{\Psi}_n^r$ is the $n \times r$ matrix with entries

$$[\mathbf{\Psi}_n^r]_{ij} = \frac{1}{\sqrt{n}} \psi_j(X_i). \tag{5.42}$$

The matrix ${\mathbf{\Psi}_n^r}^\top \mathbf{\Psi}_n^r$ contains approximate scalar products between the eigenfunctions ψ_i. Since the ψ_i are an orthogonal family of functions, it holds that ${\mathbf{\Psi}_n^r}^\top \mathbf{\Psi}_n^r \to \mathbf{I}_r$.

Furthermore, $\mathbf{K}_n^{[r]}$ is the kernel matrix construct from the kernel k^r which is obtained by truncating the expansion from Mercer's formula to the first r terms:

$$k^{[r]}(x, y) = \sum_{i=1}^{r} \lambda_i \psi_i(x) \psi_i(y). \tag{5.43}$$

The error $\|\mathbf{K}_n - \mathbf{K}_n^{[r]}\|$ is roughly as large as a constant times $\sum_{i=r+1}^\infty \lambda_i$, which is small if the eigenvalues decay quickly.

Finite-sample size bounds for the error terms $\|\mathbf{C}_n^r\|$ and $\|\mathbf{E}_n^r\|$ can be found in Chapter 3. If the eigenfunctions are uniformly bounded, $\|\mathbf{C}_n^r\|$ is treated in Theorem 3.85 while $\|\mathbf{E}_n^r\|$ is upper bounded in (3.93). For uniformly bounded kernel functions, Theorem 3.109 treats $\|\mathbf{C}_n^r\|$, and Theorem 3.131 gives estimates of $\|\mathbf{E}_n^r\|$. Finally, full relative-absolute finite sample size bounds are given in Theorem 3.94 for kernels with bounded eigenfunctions and in Theorem 3.135 for bounded kernels.

5.5 Projection Error and Finite-Sample Structure of Data in Feature Space

We now turn to the question of the effective dimension of data in feature space. We will first review this question in the context of PCA and then argue that the situation is completely different for kernel PCA. In the latter case, one can only estimate the number of leading dimensions necessary to capture most of the variance of the data, formalized by the error of projecting to the space spanned by the first d principal directions. We derive a relative-absolute bound for this projection error.

When treating vectorial data in a finite-dimensional learning problem, it often occurs that the data is not evenly distributed over all of the space but contained in a subspace of lower dimension. The reason for this is that the coordinates often correspond to certain measured features which are not completely independent but correlated in some way. For example, one coordinate might

correspond to the height of a person while another represents the weights, and the height of a person is correlated with the weight in the way that a taller person will likely weigh more.

Now since PCA computes an orthonormal basis such that the variance is maximized over the space spanned by the leading basis vectors, this means that principal values usually decay rather quickly until they reach a plateau which roughly corresponds to the noise.

To illustrate this phenomenon we consider the following model. Let $Z \in \mathbb{R}^s$ be a random variable with mean zero which models the (unobservable) features we wish to measure. The measurement process itself is modeled by a matrix $\mathbf{A} \in \mathbb{M}_{d,s}$ with $d > s$. We assume that columns of $\mathbf{A}$ are independent such that the rank of the matrix is s. The measured features $\mathbf{A}Z$ thus lie in the subspace spanned by the columns of $\mathbf{A}$ of dimension s. Finally, the measurement process is contaminated by independent additive noise $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2 \mathbf{I})$, with the variance σ_ε^2 being smaller than the variance of $\mathbf{A}Z$. The resulting measurement vector X is thus given as

$$X = \mathbf{A}Z + \varepsilon. \tag{5.44}$$

Let us compute the asymptotic principal components. It holds that

$$\mathbf{E}(XX^\top) = \mathbf{E}(\mathbf{A}ZZ^\top\mathbf{A}^\top) + \mathbf{E}(\mathbf{A}Z\varepsilon^\top) + \mathbf{E}(\varepsilon Z^\top\mathbf{A}^\top) + \mathbf{E}(\varepsilon\varepsilon^\top). \tag{5.45}$$

Since ε and Z are independent, $[\mathbf{E}(\mathbf{A}Z\varepsilon^\top)]_{ij} = \mathbf{E}([\mathbf{A}Z]_i\varepsilon_j) = \mathbf{E}([\mathbf{A}Z]_i)\mathbf{E}([\varepsilon]_j) = 0$, and $\mathbf{E}(\mathbf{A}Z\varepsilon^\top) = 0$. Likewise, $\mathbf{E}(\varepsilon Z^\top\mathbf{A}^\top) = 0$. Thus,

$$\mathbf{E}(XX^\top) = \mathbf{E}(\mathbf{A}ZZ^\top\mathbf{A}^\top) + \mathbf{E}(\varepsilon\varepsilon^\top). \tag{5.46}$$

Now note that

$$[\mathbf{E}(\varepsilon\varepsilon^\top)]_{ij} = \mathbf{E}(\varepsilon_i\varepsilon_j) = \sigma_\varepsilon^2\delta_{ij}, \tag{5.47}$$

such that

$$\mathbf{E}(XX^\top) = \mathbf{E}(\mathbf{A}ZZ^\top\mathbf{A}^\top) + \mathbf{E}(\varepsilon\varepsilon^\top) = \mathbf{E}(\mathbf{A}ZZ^\top\mathbf{A}^\top) + \sigma_\varepsilon^2\mathbf{I}_d. \tag{5.48}$$

We see that the principal values of X are the same as those of $\mathbf{A}Z$ shifted up by σ_ε^2, which means that there are $d - s$ principal values of size σ_ε^2 and then d principal values which are all larger than σ_ε^2 and correspond to the shifted principal values of the actual signal $\mathbf{A}Z$.

If we compute the PCA for a finite sample $X_1, \ldots, X_n$, due to sample fluctuations, the smallest principal value of size σ_ε^2 with multiplicity $d - s$ will be perturbed slightly, giving rise to a slope of principal values for the finite sample size PCA. Figure 5.1 plots an example. This means that the data is approximately contained in an s dimensional subspace not only in the asymptotic case, but also for finite samples. This subspace is given by the space spanned by the leading s principal components.

Now if we start with a sample $X_1, \ldots, X_n$ from some unknown source, we can calculate its PCA. The sequence of principal values then tells us something about the effective dimension of the underlying probability measure. One is usually interested in estimating the effective number of dimensions. There exist a number of approaches to do this.

The simplest approach looks for a "knee" in the sequence of eigenvalues, which is a transition into a ramp with small slope. A more sophisticated approach tests the hypothesis that the last i principal values are finite-sample approximations of those of the covariance matrix of a spherical Gaussian distribution, whose distribution can be computed in closed form.

Let us now consider the question of effective dimension when the data is mapped into a feature space. Recall that the mapping into feature space is given by (compare (5.29))

$$x \mapsto \Phi(x) = \left(\sqrt{\lambda_i}\psi_i(x)\right)_{i\in\mathbb{N}} \tag{5.49}$$

Since $(\sqrt{\lambda_i})$ is a null-sequence, we expect $\mathbf{E}([\Phi(X)]_i^2)$ to become rather small with larger i. In fact, we can compute

$$\mathbf{E}([\Phi(X)]_i^2) = \mathbf{E}(\lambda_i\psi_i^2(X)) = \lambda_i. \tag{5.50}$$

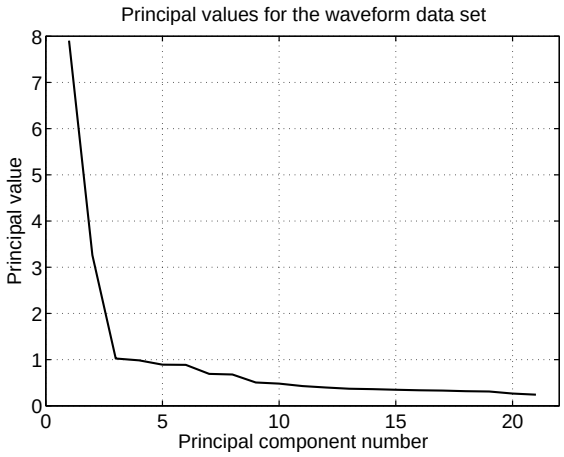


Figure 5.1: Example of the effect described in the text. The data set is the `waveform` data set from Rätsch et al. (2001). As predicted by the model, the principal values decay quickly until they pass into a slope which is due to additive noise.

Furthermore, let us compute the correlations:

$$\mathbf{E}([\Phi(X)]_i[\Phi(X)]_j) = \mathbf{E}(\sqrt{\lambda_i\lambda_j}\psi_i(X)\psi_j(X)) = \sqrt{\lambda_i\lambda_j}\langle\psi_i,\psi_j\rangle_\mu = \sqrt{\lambda_i\lambda_j}\delta_{ij}. \tag{5.51}$$

We see that $[\Phi(X)]_i$ and $[\Phi(X)]_j$ are uncorrelated for $i \neq j$. Thus the asymptotic principal directions are given by the standard unit vectors e_i in ℓ^2 (e_i having zero entries everywhere except for $[e_i]_i = 1$). Asymptotically, the principal values decay rather quickly, such that there are only a few dimensions which contain any interesting data at all.

Now coming back to the question of effective dimension, we see that the situation is completely different in the case of kernel PCA from the model discussed above, because there is no knee in the sequence of principal values, but typically these principal values just decay at a given rate. In ordinary PCA, there exists something like a background noise distributed uniformly over all directions. In kernel PCA, this noise is also mapped into the feature manifold, such that there is no slowly decreasing ramp in the sequence of principal values.

A reasonable alternative to mapping to the subspace containing the signal is to project to a number of leading dimensions such that the variance in the remaining space is negligible. The variance contained in the subspace spanned by $\psi_{d+1}, \psi_{d+2}, \ldots$ is

$$\Pi_d^2 = \sum_{i=d+1}^{\infty} \lambda_i. \tag{5.52}$$

This number will be called the *reconstruction error* (of using the subspace spanned by the first d principal directions).

For the finite sample case, we analogously define the projection error as

$$P_d^2 = \sum_{i=d+1}^{n} l_i. \tag{5.53}$$

The sum is finite in this case, because the data points completely lie in the subspace spanned by the n principal directions: From (5.36) one sees that the principal directions lie in the span of the samples $\Phi(X_i)$ in feature space. The space spanned by the $\Phi(X_i)$ has at most dimension n, and since the principal directions are orthogonal, the space spanned by those directions has dimension n, such that the data points lie in the space spanned by the first n principal directions.

By Theorem 5.39, the approximate principal values converge to the asymptotic principal values with a relative-absolute bound. Thus, the principal values will decay at roughly the same rate

until they reach the plateau defined by the limited precision of the finite precision architecture used. Therefore, we can compute the projection error for the finite case.

Theorem 5.54 (Projection error for kernel PCA) *Let k be a Mercer kernel with eigenvalues $(\lambda_i)_{i\in\mathbb{N}}$, and let $l_1, \ldots, l_n$ be the principal value estimates obtained by kernel PCA. Then, for $1 \leq r \leq n$, the finite sample size squared projection error P_d^2 is bounded by*

$$P_d^2 \leq (1 + \|\mathbf{C}_n^r\|)\Pi_d^2 + n\|\mathbf{E}_n^r\|, \tag{5.55}$$

where $\mathbf{C}_n^r$ and $\mathbf{E}_n^r$ are the same error matrices as in Theorem 5.39.

Proof By Theorem 5.39,

$$|l_i - \lambda_i| \leq \lambda_i\|\mathbf{C}_n^r\| + \|\mathbf{E}_n^r\|. \tag{5.56}$$

Therefore,

$$l_i \leq \lambda_i(1 + \|\mathbf{C}_n^r\|) + \|\mathbf{E}_n^r\|. \tag{5.57}$$

With that, we can bound P_d^2 as follows:

$$\begin{aligned} P_d^2 &= \sum_{i=d+1}^{n} l_i \leq \sum_{i=d+1}^{n} \lambda_i(1 + \|\mathbf{C}_n^r\|) + \|\mathbf{E}_n^r\| \leq \left(\sum_{i=d+1}^{n} \lambda_i\right)(1 + \|\mathbf{C}_n^r\|) + n\|\mathbf{E}_n^r\| \\ &\leq \left(\sum_{i=d+1}^{\infty} \lambda_i\right)(1 + \|\mathbf{C}_n^r\|) + n\|\mathbf{E}_n^r\| = \Pi_d^2(1 + \|\mathbf{C}_n^r\|) + n\|\mathbf{E}_n^r\|, \end{aligned} \tag{5.58}$$

which proves the bound (5.55). ■

Note that typically, $\|\mathbf{E}_n^r\|$ will be very small, roughly the size of the precision of the underlying finite-precision architecture such that the theorem states that the projection error will be a constant times the asymptotic projection error. In particular, the error decays quickly as d becomes larger, in contrast to existing bounds. (See the discussion at the end of in Section 3.4).

An important consequence of this result is that although the feature space might be infinite-dimensional, the actual data always populates only a low-dimensional subspace spanned by the first principal directions, meaning that the *effective dimensionality* of the feature space depends on the decay rate of the eigenvalues of the kernel matrix which in general depends on the smoothness of the kernel. For families of Mercer kernels like the rbf-kernels, the scale parameter thus directly controls the effective dimension of the embedding into feature space. Therefore, the sometimes expressed intuition that learning in feature space is hard due to the curse of dimensionality is not entirely correct. It is true that one has to guard against learning with arbitrarily complex functions, but in essence, the data lives in a finite-dimensional subspace of the feature space.

5.6 Conclusion

We have discussed implications of the results on eigenvalues for linear PCA and kernel PCA. Since the principal values are eigenvalues of the covariance matrix, the results from Chapter 3 could be transferred quite easily, leading to a relative perturbation result for linear finite-dimensional PCA, and a relative-absolute bound for kernel PCA. Based on this result, we were able to show that the reconstruction error becomes small when an increasing number of dimensions is used for the reconstruction, in particular for smooth kernels whose eigenvalues decay quickly. Thus, the data in feature space is contained in a finite-dimensional subspace independent of the sample size. Although the feature space is potentially infinite-dimensional, the interesting part of the data only populates a low-dimensional subspace.

Chapter 6

Signal Complexity

Abstract

We show that the basis of eigenvectors of the kernel matrix has the following property: the informative part of the label vector Y in a supervised setting is contained in the subspace spanned by the first few eigenvectors. On the other hand, the noise is distributed evenly over all of the space. This allows to separate noise from the signal. This also means that learning requires to estimate only a finite number of coefficients well, such that the whole unsupervised learning problem is effectively finite-dimensional.

6.1 Introduction

In the present chapter, we will study the relationship between the label vector and the eigenbasis of the kernel matrix in the context of supervised learning. The goal is to infer some functional dependency between object features X, which we will assume to be vectorial, and an output variable Y, which will be real (this is also called the regression setting). This dependency must be inferred given only a finite training set $(X_1, Y_1), \ldots, (X_n, Y_n)$ which is assumed to be somehow representative of the dependency one is trying to learn.

The standard model is that the X_i are i.i.d. samples from some common probability distribution μ, and the Y_i are given as samples from some fixed target function f plus some zero mean noise:

$$Y_i = f(X_i) + \varepsilon_i. \tag{6.1}$$

Now in the context of kernel methods, the question is what the relationship between the space of functions generated by the kernel functions and the data source which produces the Y is. We will address this question by studying the structure of the label vector with respect to the eigenbasis of the kernel matrix. These considerations are closely related to the results from Chapter 4 on the spectral projections of the kernel matrix.

This chapter is structured as follows. As usual, Section 6.2 briefly reviews the main results. Section 6.3 motivates the use of the eigenbasis to analyze the labels. The structure of the label vector with respect to the eigenbasis is reviewed in Section 6.4. This leads to the definition of the cut-off dimension. Two methods are proposed for estimating this cut-off dimension in Section 6.5. Section 6.6 proposes to combine the methods for estimating the cut-off dimension with rbf-kernels to perform a structural analysis of a given learning problem. Section 6.7 concludes this chapter.

6.2 Summary of Main Results

The main topic of the present chapter is the structure of a label vector given the modelling assumption (6.1). The two components of the vector, the sampled function and the additive noise,

will prove to have significantly different structure with respect to the basis of eigenvectors of the kernel matrix (we will call the basis of eigenvectors of kernel matrix the *eigenbasis* of the kernel matrix). The coefficients with respect to this basis will be called the *spectrum* of the vector. Since the eigenvectors are sorted according to their eigenvalues in non-increasing order, there is a natural ordering of the coefficients of the spectrum, with coefficients belonging to eigenvectors for larger eigenvalues coming first. When we say that the spectrum *decays*, this is meant with respect to this ordering.

Now, from the results on scalar products with eigenvectors in Chapter 4, it follows that the spectrum of a sample vector $f(\mathbf{X})$ decays quickly if f is smooth, such that the spectrum of $f(\mathbf{X})$ is contained in a number of leading coefficients. This number does not scale with the sample size n. This means that the interesting part of the label vector has a sparse representation with respect to the eigenbasis of the kernel matrix. On the other hand, the noise is evenly distributed over all of the spectrum coefficients. This means that the spectrum of the full label vector of both the sample vector of $f(\mathbf{X})$ and the noise ε will have a peculiar structure: The signal will stick out in the first few coefficients while the remainder of the spectrum consists of the noise. This leads to the definition of the *cut-off dimension*. This is a number d such that the spectrum of the sample vector dominates the leading d dimensions.

We show that this cut-off dimension can be effectively estimated by proposing two procedures, one based on a modality estimate in conjunction with resampling, and another one based on fitting a two component maximum likelihood model. On extensive simulations, it turns out that the latter method is more stable.

Finally, we propose the use of the cut-off dimension estimators to analyze the structure of the label information in a regression learning task. We show for a specific example how combining the cut-off dimension estimators with a family of kernel functions depending on a scale parameter allows to detect structure on different levels. This proves that the cut-off dimension can give an additional insight into a learning problem besides the achieved error alone.

6.3 The Eigenvectors of the Kernel Matrix and the Labels

We consider a regression setting. We wish to infer some unknown function f based on an i.i.d. sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ which forms the *training set*, where the X_i lie in the domain of f and the Y_i should somehow be representative for the images of X_i under f.

Virtually all kernel methods for regression or classification construct a fit $\hat{f}$ which can be written as follows:

$$\hat{f}(x) = \sum_{i=1}^{n} k(x, X_i)\hat{\alpha}_i + \hat{\alpha}_0, \tag{6.2}$$

where $\hat{\alpha} = (\hat{\alpha}_0, \ldots, \hat{\alpha}_n) \in \mathbb{R}^{n+1}$ is determined from the training set by the training step of the algorithm. Typical examples which include this kind of fit include Support Vector Machines of various types, Kernel Ridge Regression, and Gaussian Processes. These methods differ significantly in how $\hat{\alpha}$ is determined, but the fit has the same form. Therefore, the relationship between the space of all functions of the form (6.2) and the generating data source forms sort of an *a priori* condition of the given learning problem.

In order to specify in which sense the relationship between Y and k will be studied, let us take a look at the *in-sample fit*, which is the vector obtained by evaluating the fit function at the training points. We assume that $\hat{\alpha}_0 = 0$ which means that $f(x) \to 0$ as $\|x\| \to \infty$. The in-sample fit $\hat{Y}$ is given by

$$[\hat{Y}]_i = \hat{f}(X_i) = \sum_{j=1}^{n} k(X_i, X_j)\hat{\alpha}_j. \tag{6.3}$$

Let $\mathbf{K} \in \mathbb{M}_n$ be the matrix with entries $k(X_i, X_j)$. Then, the in-sample fit can conveniently be written in matrix form

$$\hat{Y} = \mathbf{K}\hat{\alpha}, \tag{6.4}$$

Note that this matrix $\mathbf{K}$ is equal to n times the (normalized) kernel matrix which has also been called $\mathbf{K}$ in the previous chapters. This change of notation is introduced to conform with the usual convention used in connection with kernel methods in supervised learning.

Let us re-write this formula using the spectral decomposition of $\mathbf{K} = \mathbf{U}\mathbf{L}\mathbf{U}^\top$. As usual, the columns u_i of $\mathbf{U}$ are the eigenvectors of $\mathbf{K}$, l_i the diagonal elements of $\mathbf{L}$, which are the eigenvalues of $\mathbf{K}$. Everything is sorted such that $l_1 \geq \ldots \geq l_n$. Plugging in the spectral decomposition leads to

$$\hat{Y} = \mathbf{K}\hat{\alpha} = \mathbf{U}\mathbf{L}\mathbf{U}^\top\hat{\alpha}. \tag{6.5}$$

For the individual entry this means that

$$[\hat{Y}]_i = \sum_{j=1}^{n} u_j (l_j u_j^\top \hat{\alpha}) =: \sum_{j=1}^{n} u_j \hat{\beta}_j \tag{6.6}$$

This expresses the in-sample fit as a linear combination of eigenvectors u_i of $\mathbf{K}$. This way of decomposing the fit $\hat{Y}$ has several advantages compared with the plain in-sample fit formula (6.3), where the fit is constructed from the columns of the kernel matrix.

First of all, the eigenvectors form an orthonormal set of vectors. Therefore, the weights $\hat{\beta}_j$ control orthogonal components, which can be thought of as geometrically independent components. Furthermore, the eigenvectors usually increase in complexity as their associated eigenvalue becomes smaller (Williams and Seeger, 2000). Therefore, (6.6) decomposes $\hat{Y}$ with respect to components with increasing complexity. These are only preliminary considerations. As will be shown in this chapter, the most important reason is that with respect to the basis of eigenvectors of the kernel matrix, smooth and noisy parts of the label information have significantly different structure. It will turn out that a smooth function will have a sparse representation in this basis, while i.i.d. noise will have evenly distributed coefficients. These distinctions allow us to estimate the number of relevant dimensions in feature space.

Throughout this chapter, we will make the following *modelling assumption*. We assume that Y_i is given by

$$Y_i = f(X_i) + \varepsilon_i, \qquad (\varepsilon_1, \ldots, \varepsilon_n)^\top \sim \mathcal{N}(0, \sigma_\varepsilon^2 \mathbf{I}_n), \tag{6.7}$$

which means that the Y_i are the sampled values of f plus additive zero mean independent Gaussian noise. We will call

$$Y = (Y_1, \ldots, Y_n)^\top \in \mathbb{R}^n \tag{6.8}$$

the *label vector*.

Moreover, we will make certain smoothness assumptions on f. We will assume that f lies in the image of the operator T_k. Let λ_i be the eigenvalues of T_k and ψ_i its eigenfunctions. Then, $f \in \operatorname{ran} T_k$ is equivalent to the existence of a sequence $(\alpha_i)_{i\in\mathbb{N}} \in \ell^2$ such that

$$f = \sum_{i=1}^{\infty} \alpha_i \lambda_i \psi_i. \tag{6.9}$$

For smooth kernels like the rbf-kernel, the operator T_k acts as a smoother such that this ensures a certain degree of regularity. One can show (compare Lemma 2.33) that the amount of regularity directly corresponds to the norm of the parameter sequence (α_i).

6.4 The Spectrum of the Label Vector

We now introduce the notion of the *spectrum of the label vector*, which is the vector of coefficients of Y with respect to the eigenbasis of Y. Since the kernel matrix $\mathbf{K}$ is symmetric, its eigenvectors $u_1, \ldots, u_n$ are orthogonal and if they are normalized to unit length, $\mathcal{U} = \{u_1, \ldots, u_n\}$ forms an orthonormal basis of $\mathbb{R}^n$. Therefore, computing the coefficient of Y with respect to the eigenbasis

of $\mathbf{K}$ is particularly simple, because one only has to compute the scalar products of Y with the eigenvectors:

$$[s]_i = u_i^\top Y. \tag{6.10}$$

In matrix notation, the vector s of all coefficients can conveniently be written as

$$s = \mathbf{U}^\top Y. \tag{6.11}$$

In analogy to the term *eigenbasis*, we will call s the *eigencoefficients* of Y with respect to the kernel matrix $\mathbf{K}$.

We will also call s the *spectrum* of Y. This term is motivated by the observation that the eigenvectors of smooth kernels (for example, rbf-kernels (2.17)) typically look like sine waves whose complexity increases as the eigenvalue becomes smaller. For data in a vector space with dimension larger than one, similar observations can be made (see for example the paper by Schölkopf et al. (1999)). Unfortunately, since the interplay between the underlying distribution μ of the data and the kernel function is not yet completely understood although first steps have been made (Williams and Seeger, 2000), this observation cannot be put in more rigorous terms. In summary, typically, computing the eigencoefficients of a vector decomposes the vector into orthogonal components with increasing complexity, not unlike a Fourier transformation. Therefore, in analogy to the Fourier transformation, we will call the eigencoefficients of Y the spectrum of Y. We will see below that this terminology is actually supported by observations concerning the structure of the signal and noise part of the label vector.

By the modelling assumption (6.7), it holds that $Y_i = f(X_i) + \varepsilon_i$. Let us write $f(\mathbf{X}) = (f(X_1), \ldots, f(X_n))^\top$ and $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^\top$. Then, the spectrum of Y is

$$\mathbf{U}^\top Y = \mathbf{U}^\top (f(\mathbf{X}) + \varepsilon) = \mathbf{U}^\top f(\mathbf{X}) + \mathbf{U}^\top \varepsilon. \tag{6.12}$$

Thus, the spectrum of Y is a superposition of the spectrum of the sample vector of f and of the noise ε. As we will see, $\mathbf{U}^\top f(\mathbf{X})$ and $\mathbf{U}^\top \varepsilon$ have significantly different structures.

6.4.1 The Spectrum of a Smooth Function

Recall that we assumed that f was smooth in the sense that $f = \sum_{\ell=1}^{\infty} \alpha_\ell \lambda_\ell \psi_\ell$ for some sequence $(\alpha_\ell) \in \ell^2$. By the general results cited in Section 4.4, we know that the scalar products

$$\frac{1}{\sqrt{n}} |u_i^\top f(\mathbf{X})| \tag{6.13}$$

converge to $|\langle \psi_i, f \rangle_\mu|$ (taking the necessary precautions for eigenvalues with multiplicity larger than 1). Since the (ψ_i) form an orthonormal family of functions, it holds that

$$\langle \psi_i, f \rangle_\mu = \alpha_i \lambda_i. \tag{6.14}$$

The asymptotic (infinite) spectrum of f given by the sequence $(\langle \psi_i, f \rangle_\mu)_{i \in \mathbb{N}}$ decays rather quickly, because $(\alpha_i) \in \ell^2$ and $(\lambda_i) \in \ell^1$. In particular, for every $\varepsilon > 0$, there exist only a finite number of entries which are larger than ε. In other words, by only considering a finite number of basis functions $\psi_1, \ldots, \psi_r$, we can already reconstruct f up to a small error. More specifically, for any error $\varepsilon > 0$, a finite reconstruction can be found whose error does not exceed ε. In other words, f has finite complexity at any given scale.

We are interested in the question whether the sampled spectrum $s = \mathbf{U}^\top f(\mathbf{X})$ has similar properties. In Chapter 4, we derived a relative-absolute envelope on $|u_i^\top f(\mathbf{X})|/\sqrt{n}$. This is an upper bound which does not converge to zero as the number of samples goes to infinity, but which is nevertheless quite small for certain indices i. By Theorem 4.92, we know that

$$\frac{1}{\sqrt{n}} |u_i^\top f(\mathbf{X})| \leq l_i O\left(\sum_{\ell=1}^{r} |\alpha_\ell| \right) + \varepsilon(r, f), \tag{6.15}$$

where r can be chosen such that $\varepsilon(r, f)$ becomes very small. This means that the coefficients of s also decay quickly and that the sample vector $f(\mathbf{X})$ will be contained in the space spanned by the first few eigenvectors of $\mathbf{K}_n$, and that this number is independent of the sample size.

In summary, a smooth function will also have a quickly decaying spectrum on a finite sample. There will only be a certain number of eigencoefficients which are large.

Note the similarity to the observations from the last Chapter. In Section 5.5, we showed that a finite sample will be contained in a low-dimensional subspace in feature space. In this section we have shown that the finite-dimensional version of the label vector is essentially contained in the first few dimension with respect to the eigenbasis of the kernel vector.

6.4.2 The Spectrum of Noise

Recall that we assumed that $\varepsilon = \mathcal{N}(0, \sigma_\varepsilon^2 \mathbf{I})$. Let us study some stochastic properties of $\mathbf{U}^\top \varepsilon$. First of all note that $\mathbf{E}(\mathbf{U}^\top \varepsilon) = 0$, such that the noise has mean zero also with respect to the eigenbasis of $\mathbf{K}$.

The eigenbasis $u_1, \ldots, u_n$ of $\mathbf{K}$ depends only on the X_i which are independent of ε. Therefore, $x \mapsto \mathbf{U}^\top x$ computes a random rotation of x which is independent of the realization of ε. Furthermore ε is a spherical distribution, such that $\mathbf{U}^\top \varepsilon$ is just a rotated version of ε which still has the same distribution:

$$\mathbf{U}^\top \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2 \mathbf{I}). \tag{6.16}$$

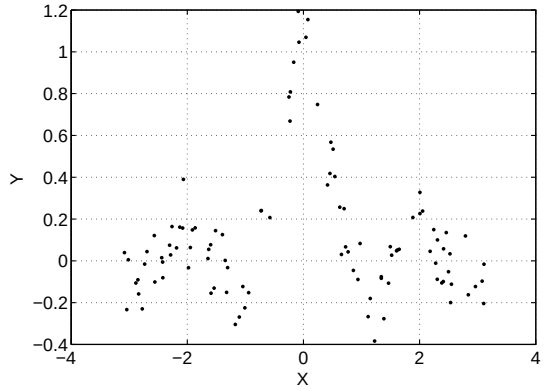
Therefore, the spectrum of ε will be more or less *flat*, which means that it is equally distributed over all components, and the components are typically neither very small nor very large. This will be still be true to a lesser extent, if ε is not spherically distributed. In order for ε to be concentrated along one of the eigenvector directions of $\mathbf{K}$, the noise has to have a shape similar to the eigenvectors, but in any case, the eigenvectors will be independent of ε. Since the eigenvectors to larger eigenvalues are more or less smooth, this is rather unlikely if the noise has zero mean and is independent while not being identically distributed. The smaller eigenvectors are not very smooth but also not very stable, such that again, the scalar products will be more or less random and the spectrum will be flat.

This characterization of the shape of the spectrum is reminiscent of the question of effective dimensions in the context of PCA from Section 5.5, although the objects involved and the underlying mechanics should not be confused: In PCA, the eigenvalues of the covariance matrix are considered, whereas here, we are looking at the scalar products between the vector of all labels of the training set and the eigenbasis of the kernel matrix. Furthermore, the present characterization depends crucially on the relative-absolute envelope for scalar products with eigenvectors of the kernel matrix which form an original contribution of this thesis. Even the asymptotic results from Koltchinskii (1998) date back only a few years, whereas the PCA setting is known at least since (Schmidt, 1986).

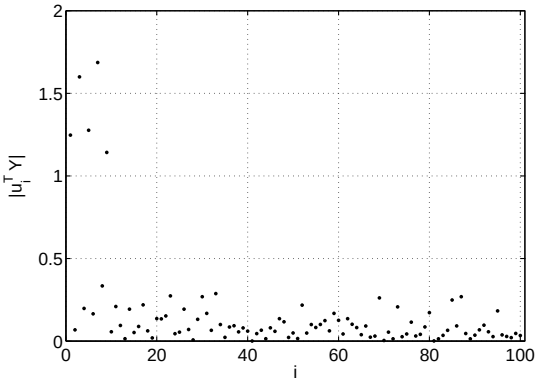
6.4.3 An example

Let us take a look at an example. As usual, we take the noisy sinc function and the rbf-kernel with kernel width $w = 1$. Figure 6.1 plots the raw data, and the spectra of Y and its components $f(\mathbf{X})$ and ε. As predicted, the spectrum of the sample vector $f(\mathbf{X})$ decays quickly (note the logarithmic scale!), while the spectrum of the noise is flat. We also see how the spectrum of the label vector sticks out of the noise. This is due to the fact that the whole variance of the label vector has to be contained in a few dimensions, while the variance of the noise can be spread out evenly. Thus, even if $\|f(\mathbf{X})\| = \|\varepsilon\|$, the spectrum of $f(\mathbf{X})$ will stick out of the noise spectrum.

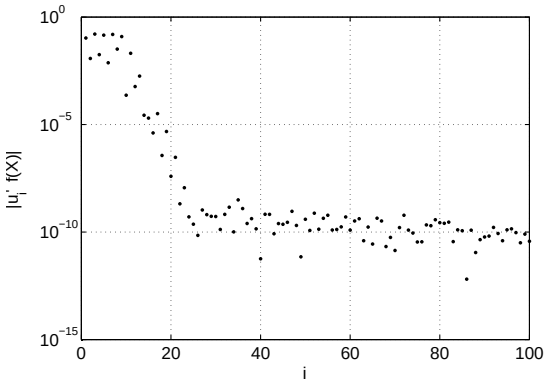
It is instructive to compare the spectrum of $f(\mathbf{X})$ from Figure 6.1 with that of a function which is not smooth in the sense that it lies in the range of T_k. For continuous kernels, an example for such a function is the sign function which is discontinuous at 0. Figure 6.2 plots the spectrum of



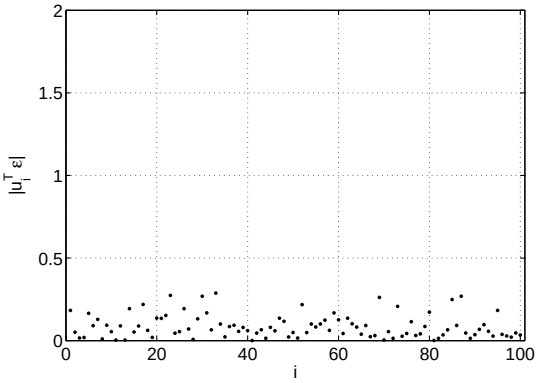
(a) A typical data set.



(b) The spectrum of the label vector Y.



(c) The spectrum of the sample vector $f(\mathbf{X})$ (in log-scale).



(d) The spectrum of the noise ε.

Figure 6.1: Spectrum for the noisy sinc example. The spectrum of the sample vector $f(\mathbf{X})$ decays rapidly whereas the spectrum of the noise vector ε is flat.

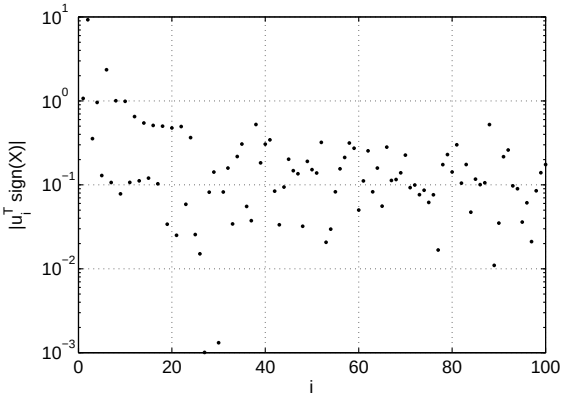


Figure 6.2: Spectrum of the sign function for values uniformly drawn in $[-\pi, \pi]$ with respect to the rbf-kernel. Since the sign function is discontinuous and does therefore not fulfill the regularity assumption, the spectrum decays much slower than in the noisy sinc function example.

the sign function. We see that the spectrum decays, but much more slowly than that in Figure 6.1. Therefore, the smoothness condition is crucial to obtain rapid decay.

6.4.4 The Cut-off Dimension

Let us summarize the above observations. By the convergence results in Chapter 4, we know that the spectrum of the sample vector of a smooth function has only a finite number of large entries (which is moreover independent of n), whereas the spectrum of independent noise is evenly distributed over all coefficients. The eigenbasis of the kernel matrix yields a representation of the label vector in which the interesting part in the label vector and the noise have significantly different structures.

These considerations lead to the definition of the *cut-off dimension* for a given label vector Y:

Definition 6.17 Given a label vector $Y = f(\mathbf{X}) + \varepsilon$ of length n, the *cut-off dimension* is the largest number $1 \leq d \leq n$, such that

$$|[\mathbf{U}^\top f(\mathbf{X})]_d| > |[\mathbf{U}^\top \varepsilon]_d|. \tag{6.18}$$

Thus, the cut-off dimension is the size of the part of the spectrum of $f(\mathbf{X})$ which sticks out of the noise. For the example, from Figure 6.1(b), the cut-off dimension seems to be 9. From Figure 6.1(c), we see that from that point onwards the spectrum is smaller than 10^{-2} which is relatively small, but still a bit away from the true residual at around $d = 20$. In the presence of noise, this portion of $f(\mathbf{X})$ is not visible. The following lemma summarizes the intuition that the cut-off dimension captures $f(\mathbf{X})$ up to the noise level.

Lemma 6.19 *Let $Y = f(\mathbf{X}) + \varepsilon$ and d be the cut-off dimension. Let $\pi_d x = \sum_{i=1}^{d} u_i u_i^\top x$ be the projection of $x \in \mathbb{R}^n$ to the space spanned by the first d eigenvectors of $\mathbf{K}$. Then,*

$$\frac{1}{n}\|f(\mathbf{X}) - \pi_d f(\mathbf{X})\|^2 \leq \frac{1}{n}\|\varepsilon\|^2 \to_{a.s.} \mathsf{Var}(\varepsilon), \tag{6.20}$$

where the limit is taken by letting the number of samples n (and $\mathbf{X}$ accordingly) tend to infinity.

Proof It holds that $f(\mathbf{X}) - \pi_d f(\mathbf{X}) = \sum_{i=d+1}^{n} u_i u_i^\top f(\mathbf{X})$. Therefore,

$$\begin{aligned}
\frac{1}{n}\|f(\mathbf{X}) - \pi_d f(\mathbf{x})\|^2 = \frac{1}{n}\sum_{i=d+1}^{n} \|u_i u_i^\top f(\mathbf{X})\|^2 &= \frac{1}{n}\sum_{i=d+1}^{n} \left(u_i^\top f(\mathbf{X})\right)^2 \\
&\overset{(1)}{\leq} \frac{1}{n}\sum_{i=d+1}^{n} (u_i^\top \varepsilon)^2 \leq \frac{1}{n}\sum_{i=1}^{n} (u_i^\top \varepsilon) = \frac{1}{n}\|\mathbf{U}^\top \varepsilon\|^2 \\
&\overset{(2)}{=} \frac{1}{n}\|\varepsilon\|^2 \overset{(3)}{\to_{\text{a.s.}}} \mathsf{Var}(\varepsilon_1) = \sigma_\varepsilon^2.
\end{aligned} \tag{6.21}$$

where (1) follows from the definition of the cut-off dimension, (2) follows because $\mathbf{U}$ is an orthogonal matrix, and (3) from the strong law of large numbers. Note that although d is random, the upper bound holds nevertheless, because in step (3) where the limit is taken, d has already been eliminated. ■

In words, if one considers only the reconstruction of $f(\mathbf{X})$ up to the cut-off dimension, the reconstruction error is asymptotically smaller than the noise variance.

Note that we have neglected the fact that the spectrum of $f(\mathbf{X})$ decays quickly. Therefore, the actual error will be much smaller than predicted by the last lemma. On the positive side, this lemma will also hold for non-smooth function f. In fact, it holds for any vectors $f(\mathbf{X})$.

6.4.5 Connections to Wavelet Shrinkage

Wavelet shrinkage is a spatially adaptive technique for learning a function when the sample points X_i are given as equidistant points. Such data sets typically occur in signal or image processing. For such point sets, one can define a wavelet basis which leads to a multi-scale decomposition of the signal. Wavelet shrinkage then proceeds by selectively thresholding the wavelet coefficients of the signal, resulting in a reconstruction of the original signal which is able to recover both, smooth areas and jump discontinuities. In this respect, wavelet methods often show superior performance, in particular compared to linear methods. It has even been shown by Donoho and Johnstone (1998) that using certain thresholding schemes, the resulting method is nearly minimax optimal over any member of a wide range of so-called Triebel and Besov-type smoothness classes, and also asymptotically minimax optimal over certain Besov bodies.

The connection to the discussion here is given by the fact that wavelet shrinkage is analyzed in terms of the so-called *sequence space*. This space is obtained by considering the wavelet coefficients, just as we have considered the coefficients with respect to the eigenvector of the kernel matrix. In both cases, the coefficients represent the original label vector Y with respect to an orthonormal basis. As has been discussed above, after the basis transformation, the noise stays normally distributed.

Now interestingly, using the wavelet analysis, a noiseless signal will typically have only a small number of large wavelet coefficients, while the remaining coefficients will be rather small. On the other hand, as explained above, the noise will contribute evenly to all wavelet coefficients. Based on the theoretical results from Chapter 4, we are now in a position to state that essentially the same condition holds in the case of the eigenvector basis of the kernel matrix. Now, while the eigenvectors of the kernel matrix typically do not lead to a multi-scale decomposition of the label vector, on the other hand, kernel methods naturally extend to non-equidistant sample points, a setting where application of wavelet techniques is not straight-forward.

Below, when we discuss practical methods for estimating the cut-off dimension, we will return to the topic of wavelet shrinkage and discuss the applicability of methods like VisuShrink and SureShrink (introduced in (Donoho and Johnstone, 1995) and (Donoho et al., 1995)) for determining the threshold coefficients in the kernel setting.

6.5 Estimating the Cut-off Dimension given Label Information

Given the spectrum s of a label vector as in (6.11), we have to solve the task of estimating a cut-off dimension d, such that the signal is contained in the space spanned by the first d eigenspaces of the kernel matrix $\mathbf{K}$. In this section, we will propose two procedures for estimating the cut-off dimension. We also discuss the connection of thresholding methods from the framework of wavelet shrinkage.

6.5.1 Resampling Based Estimate of Modality

We have to distinguish components of the spectrum $s = \mathbf{U}^\top Y$ which merely reflect the noise from those which are due to some structure in the data. The difference is that given the modelling assumption, the noise coefficients will have roughly a unimodal distribution with mean 0, while the coefficients will have some distribution with a mean different from zero in the other case. Moreover, the signs of the scalar products are rather arbitrary because both u and $-u$ are valid eigenvectors. Therefore, interesting coefficients will have a bimodal distribution as if a random variable with a unimodal distribution has been multiplied by a random sign. Given a learning problem, we only have one label vector Y. In order to simulate more than one sample, we use a resampling scheme, picking a subset of the training examples to construct a kernel matrix and computing the eigencoefficients. Thus, the method we will introduce in this section combines resampling with a test for bimodality based on kernel density estimates to distinguish signal coefficients from noise coefficients.

Let us first discuss a how to detect bimodality. Let X be a real random with differentiable density $p(x)$. If p is sufficiently smooth, unimodality of X can be defined as p having only a single maximum, such that there exists only one zero of p'. If X is bimodal, p' has three zeros, one for the peaks of the two modes and one for the valley in between. These definitions hold for example for two well-separated normal distributions. For more general distributions, these definitions may fail, but note that for such distributions the notion of modality is also not well-defined.

Now assume that instead of p, only a finite number of i.i.d. samples $X_1, \ldots, X_n$ is given. We want to estimate a sufficiently smooth density given the $X_1, \ldots, X_n$. This can be accomplished by using kernel density estimates (see for example (Duda et al., 2001, Section 4.3)). We use estimates based on Gaussian kernels. The estimated density is then:

$$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^{n} g_{\sigma^2}(x - X_i), \qquad \text{with } g_{\sigma^2}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right), \tag{6.22}$$

where σ^2 is a width parameter. Smaller σ^2 leads to finer estimates, while larger σ^2 lead to coarser estimates.

We use the following heuristic for choosing σ^2: Let $[a, b]$ be the smallest interval which contains $X_1, \ldots, X_n$. Then, let $r = b - a$, and set

$$\sigma = \frac{2r}{\sqrt{n}}. \tag{6.23}$$

It is known (Duda et al., 2001) that this choice of scaling with n guarantees that convergence of the density to the true density. Let $\hat{p}_i = \hat{p}(X_i)$ for $1 \leq i \leq n$.

We estimate the derivatives of p' at the points X_i by the differences

$$\hat{p}'_i = \frac{\hat{p}_{i+1} - \hat{p}_i}{X_{i+1} - X_i}, \qquad \text{for } 1 \leq i \leq n-1, \tag{6.24}$$

where we have assumed that the X_i have been sorted in ascending order.

Finally, the number of modes can be determined by the number of *zero crossings* c, which are the number of times $\hat{p}'_i$ has a different sign than $\hat{p}'_{i+1}$. The number of modes is then estimated as $(c+1)/2$.

Finally, one has to guard the algorithm from the case where the overall variance of X is so small that numerical problems might arise. Therefore, we check beforehand if the estimated variance of $X_1, \ldots, X_n \leq 10^{-14}$. Then, X is estimated to be unimodal. The complete algorithm is summarized in Figure 6.3.

We have also tried replacing the heuristic for choosing the kernel width by likelihood cross-validation, but although asymptotic optimality has been proved recently (van der Laan et al., 2004), the estimates were in general not smooth enough to allow for a modality estimate based on the zeros of the differential of the kernel density estimate.

In order to simulate a sample from $\mathbf{U}^\top Y$ for different realizations of X and Y, we resample the given label vector Y and object samples $X_1, \ldots, X_n$ by picking R random samples with replacement (actually, without replacement works just as well). Let $i_1, \ldots, i_R$ be the chosen indices. Then, define

$$X'_j = X_{i_j}, Y'_j = Y_{i_j} \text{ and } Y' = (Y'_1, \ldots, Y'_R). \tag{6.25}$$

for $1 \leq j \leq R$. Based on these, set up a kernel matrix $\mathbf{K}'$ with entries $k(X'_i, X'_j)$, compute its eigendecomposition $\mathbf{U}'\mathbf{L}'\mathbf{U}'^\top$ and the resampled spectrum vector $s' = \mathbf{U}'^\top Y'$.

This process is repeated I times which generates I samples for each coordinate of s'. Based on these samples, the number of modes is estimated for each coordinate. Additionally, for each coordinate, we extend the values obtained by the resampling by a copy multiplied by -1 to ensure that the random change of sign is evenly distributed. The cut-off dimension is then the last index such that the associated coordinate is bimodal. The whole algorithm is summarized in Figure 6.4.

Estimating the number of modes

Input: real numbers $X_1, \ldots, X_n$
Output: number of modes m

0 if estimated variance of $X_1, \ldots, X_n < 10^{-14}$, return $m \leftarrow 1$
1 sort $X_1, \ldots, X_n$.
compute range of data
2 set $a \leftarrow \min(X_1, \ldots, X_n)$, $b \leftarrow \max(X_1, \ldots, X_n)$, and $r \leftarrow b - a$
compute kernel density estimate
3a set $\sigma^2 \leftarrow 4r^2/n$
for $1 \le i \le n$,
3b $$\hat{p}_i \leftarrow \frac{1}{n} \sum_{j=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{|X_i - X_j|^2}{2\sigma^2}\right).$$
compute discrete derivative of $\hat{p}$
4 for $1 \le i \le n-1$,
$$\hat{p}'_i \leftarrow \frac{\hat{p}_{i+1} - \hat{p}_i}{X_{i+1} - X_i}.$$
5 count zero crossings c of $\hat{p}'_1, \ldots \hat{p}'_{n-1}$.
6 return $m \leftarrow (c+1)/2$

Figure 6.3: Estimating the number of modes.

Estimating the cut-off dimension by a modality estimate.

Input: vectors $X_1, \ldots, X_n \in \mathbb{R}^d$
label vector $Y = (Y_1, \ldots, Y_n) \in \mathbb{R}^n$.
kernel function k
Parameters: size of resample R
number of resample iterations I
Output: cut-off dimension $d \in \{1, \ldots, R\}$.

let $\mathbf{S}$ be an $I \times R$ matrix
Compute resamples
1 for $1 \le i \le I$
1a let $i_1, \ldots, i_R$ be uniformly drawn integers in $\{1, \ldots, n\}$.
1b let $\mathbf{K}'$ be the kernel matrix based on $X_{i_1}, \ldots X_{i_R}$,
1c compute the eigendecomposition of $\mathbf{K}' = \mathbf{U}'\mathbf{L}'\mathbf{U}'^{\top}$.
1d let $Y' \leftarrow (Y_{i_1}, \ldots, Y_{i_R})$
1e set $[\mathbf{S}]_{ir} \leftarrow [\mathbf{U}'^{\top} Y']_r$ for $1 \le r \le R$
estimate number of modes
2 for $1 \le r \le R$,
compute the number of modes of m_r of the vector $([\mathbf{S}]_{ir})_i$ using Algorithm 6.3
3 return the last index d such that $m_d > 1$.

Figure 6.4: Estimating the cut-off dimension by resampling and a modality estimate.

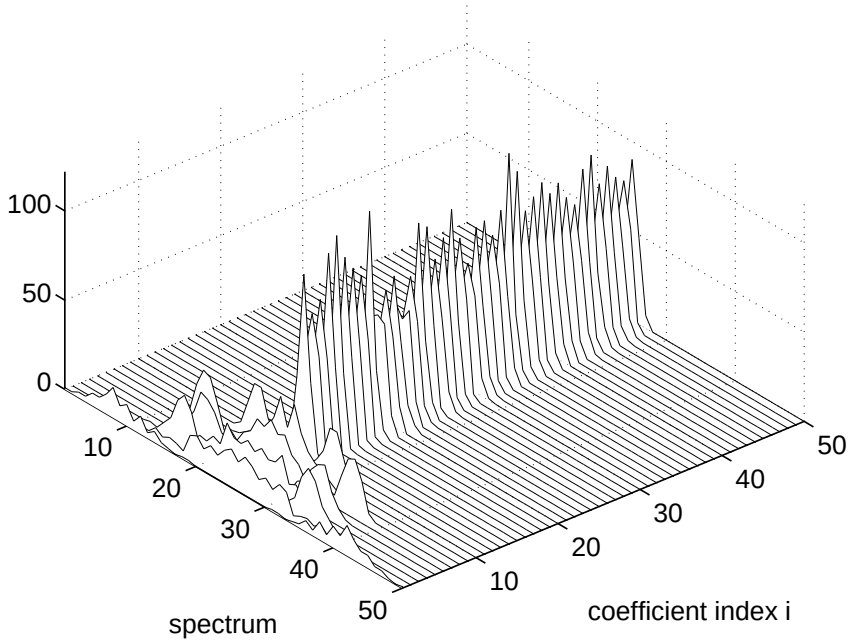


Figure 6.5: Histograms of the eigencoefficients after resampling for the noisy sinc function example. One can see that among the first 9 directions there are directions which have a clearly bimodal distribution.

The algorithm is straightforward maybe with the exception that $\mathbf{S}$ is filled in row-wise but analyzed column-wise. The reason is that in Step 1e, the spectrum for a resample Y' is computed, but the number of modes is calculated for each coordinate, not each resample.

Figure 6.5 shows histograms for the different entries of the spectrum based on the resampled data. We can see that the uneven coefficients 1 to 9 in fact have a bimodal distribution.

6.5.2 Two-Component Model

The main drawback of the algorithm from the last section is that it is computationally very expensive for small sample sizes (for large sample size, it might even perform better by considering only small subsamples. Then again, one can always only consider a subsample of the whole set to speed up the estimation of the cut-off dimension). We therefore propose an alternative algorithm which appears to be less principled but which will prove to work very well.

The basic idea is that the spectrum consists of two parts which have different variance. The model assumes that the coefficients of the spectrum $s = (s_1, \ldots, s_n)^\top$ are distributed as

$$s_i \sim \begin{cases} \mathcal{N}(0, \sigma_1^2) & 1 \leq i \leq d \\ \mathcal{N}(0, \sigma_2^2) & d+1 \leq i \leq n. \end{cases} \tag{6.26}$$

Under the basic modelling assumptions from (6.7), this model is actually justified for the second part reflecting only the noise in the data. For the first part, a Gaussian distribution has been chosen as a default distribution, since no further prior information is available.

In order to estimate the cut-off dimensions, we perform a maximum likelihood fit of the model parameters $(d, \sigma_1^2, \sigma_2^2)$. For a fixed d, the variances are estimated by

$$\sigma_1^2 = \frac{1}{d} \sum_{i=1}^{d} s_i^2, \qquad \sigma_2^2 = \frac{1}{n-d} \sum_{i=d+1}^{n} s_i^2. \tag{6.27}$$

We have to compute the negative log-likelihood for different values of d. The cut-off dimension will then be the value d which minimizes the negative log-likelihood:

$$-\log(p(s_1,\ldots,s_n)) = -\log\left(\prod_{i=1}^{d} p(s_i)\prod_{i=d+1}^{n} p(s_i)\right) = -\sum_{i=1}^{d}\log p(s_i) - \sum_{i=d+1}^{n}\log p(s_i). \quad (6.28)$$

We consider the first sum first:

$$-\sum_{i=1}^{d}\log p(s_i) = \sum_{i=1}^{d}\left(\frac{1}{2}\log(2\pi\sigma_1^2) + \frac{1}{2\sigma_1^2}s_i^2\right) = \frac{d}{2}\log(2\pi\sigma_1^2) + \frac{1}{2\sigma_1^2}\sum_{i=1}^{d} s_i^2. \quad (6.29)$$

Recall the definition of the estimate σ_1^2 from (6.27). Therefore,

$$-\sum_{i=1}^{d}\log p(s_i) = \frac{d}{2}\log(2\pi\sigma_1^2) + \frac{d}{2}. \quad (6.30)$$

Analogously, we obtain that

$$-\sum_{i=d+1}^{n}\log p(s_i) = \frac{n-d}{2}\log(2\pi\sigma_2^2) + \frac{n-d}{2}. \quad (6.31)$$

Therefore,

$$\begin{aligned}(6.28) &= \frac{d}{2}\log(2\pi\sigma_1^2) + \frac{n-d}{2}\log(2\pi\sigma_2^2) + \frac{d}{2} + \frac{n-d}{2}\\ &= \frac{1}{2}\big(d\log(\sigma_1^2) + (n-d)\log(\sigma_2^2) + (d+n-d)\log(2\pi) + d + n - d\big) \quad (6.32)\\ &= \frac{1}{2}\big(d\log(\sigma_1^2) + (n-d)\log(\sigma_2^2) + n(\log(2\pi)+1)\big)\end{aligned}$$

Since we are only interested in the argument of the maximum with respect to d, we can omit the factor $1/2$ and the term $n(\log(2\pi)+1)$. Therefore, the estimated cut-off dimension is

$$\hat{d} = \operatorname*{argmax}_{1\le d\le n-1}\big(d\log(\sigma_1^2) + (n-d)\log(\sigma_2^2)\big) \quad (6.33)$$

with σ_1^2, σ_2^2 defined in (6.27).

For practical purposes, it is often advisable to limit the candidate values d for estimated cut-off dimension $\hat{d}$, because small sample size fluctuations can otherwise lead to likelihoods becoming very large for d close to n. Therefore, we suggest limiting d to $\lceil n/2\rceil$ which has proven to be sufficient for all applications discussed in this thesis. Figure 6.6 summarizes the algorithm.

Let us finally look at an example. We again take the noisy sinc function example. In Figure 6.7, we again see the spectrum from Figure 6.1, and next to it the negative log-likelihoods computed by Algorithm 6.6. The minimum is at $\hat{d}$ which nicely coincides with our observations when discussing the cut-off dimension for this example in Section 6.4.4 .

6.5.3 Threshold Estimation via Wavelet Shrinkage Methods

As discussed in Section 6.4.5, there is a considerable similarity between the notion of estimation in sequence space from the framework of wavelet shrinkage methods, and the spectrum of a label vector derived in this chapter.

The problem of estimation in sequence space is define as follows (see, for example, Donoho et al. (1995)): Suppose we observe sequence data

$$s_i = \theta_i + \varepsilon_i, \qquad i \in I, \quad (6.34)$$

Estimating the cut-off dimension by the two-component model

Input: kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$,
labels $Y = (Y_1, \ldots, Y_n)^\top \in \mathbb{R}^n$.

Output: estimated cut-off dimension $\hat{d} \in \{2, \ldots, n-1\}$

1 compute eigendecomposition $\mathbf{K} = \mathbf{ULU}^\top$ with $\mathbf{L} = \mathrm{diag}(l_1, \ldots, l_n)$, $l_1 \geq \ldots \geq l_n$.

2 $s \leftarrow \mathbf{U}^T Y$.

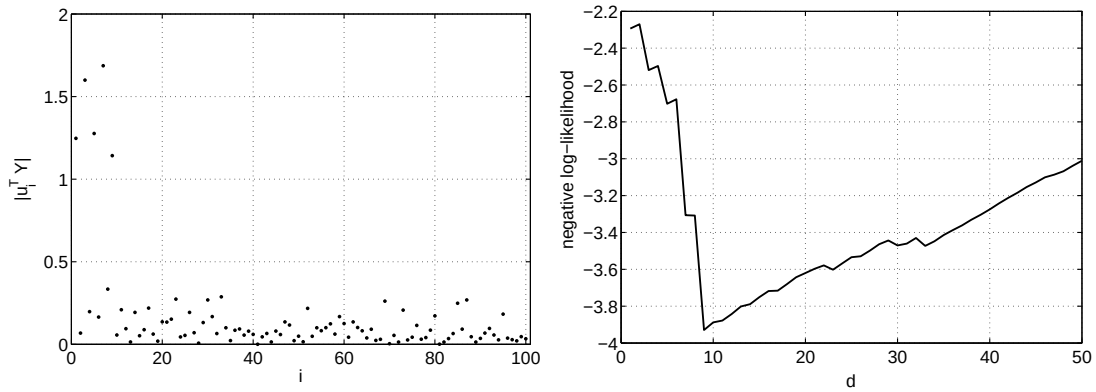
3 for $j = 2, \ldots, \lceil n/2 \rceil$,

3a $$\sigma_1^2 \leftarrow \frac{1}{j} \sum_{i=1}^{j} s_i^2, \quad \sigma_2^2 \leftarrow \frac{1}{n-j} \sum_{i=j+1}^{n} s_i^2,$$

3b $$l_j \leftarrow \frac{j}{n} \log \sigma_1^2 + \frac{n-j}{n} \log \sigma_2^2.$$

4 return $\hat{d} \leftarrow \underset{j=1,\ldots,\lceil n/2 \rceil}{\operatorname{argmin}} l_j$

Figure 6.6: Estimating the cut-off dimension given a kernel matrix and a label vector.



(a) The spectrum of the label vector Y

Figure 6.7: The negative log-likelihood for the noisy sinc function example.

where I is some index set, $\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2)$, and $\theta = (\theta_i)$ is unknown. The goal is to estimate θ with small squared error $\|\hat{\theta} - \theta\|^2 = \sum_{i \in I} (\hat{\theta}_i - \theta_i)^2$.

For wavelet shrinkage, one usually considers some form of thresholding in order to estimate $\hat{\theta}_i$, for example *hard-thresholding*

$$\hat{\theta}_i^H = \begin{cases} s_i & |s_i| > t, \\ 0 & |s_i| \leq t, \end{cases} \tag{6.35}$$

and *soft-thresholding*

$$\hat{\theta}_i^S = \operatorname{sign}(s_i)(s_i - t)_+, \tag{6.36}$$

with $(x)_+ = \max(x, 0)$. One usually prefers soft-thresholding, for example, because it is continuous.

We discuss two standard methods for estimating the threshold for soft-thresholding: VisuShrink and SureShrink. In both cases, we assume that the noise ε_i has variance one.

VisuShrink is based on the observation that the maximum of n normally distributed random variables can be bounded by $\sqrt{2 \log n}$ (Lepskii, 1990). More concretely, let $X_1, \ldots, X_n$ be i.i.d. $\mathcal{N}(0, 1)$ variables. Then,

$$\mathbf{P}\left\{ \sup_{1 \leq i \leq n} |X_i| \leq \sqrt{2 \log n} \right\} \to 1 \tag{6.37}$$

as $n \to \infty$. Thus, one sets $t = \sqrt{2 \log n}$.

SureShrink is based on SURE, Stein's unbiased risk estimator (Donoho and Johnstone, 1995). Let $\mu \in \mathbb{R}^n$, and let $X \sim \mathcal{N}(\mu, \mathbf{I})$. Then, let $\hat{\mu}$ be an estimator for μ. In Stein (1981), a method was developed which allows to estimate the loss $\|\hat{\mu} - \mu\|^2$ in an unbiased fashion. This is done as follows: Assume that $\hat{\mu}(X) = X + g(X)$, where $g \colon \mathbb{R}^n \to \mathbb{R}^n$, and g is weakly differentiable. Then,

$$\mathbf{E}(\|\hat{\mu}(X) - \mu\|^2) = N + \mathbf{E}\left(\|g(X)\|^2 + 2 \sum_{i=1}^{N} \frac{\partial}{\partial X_i} [g(X)]_i \right). \tag{6.38}$$

This means that the expression on the right hand side without the expectation is an unbiased estimator of the true risk.

Now in the context of estimation in sequence space, due to the fact that the noise is normally distributed, the problem of estimating θ is that of estimation the mean of a multivariate normal distribution. For the soft-thresholding procedure from (6.36), set $\hat{\theta}_i^S = s_i + \theta_i^S - s_i$. Therefore, we have $[g(s)]_i = \theta_i^S - s_i$. We compute

$$[g(s)]_i = \begin{cases} -s_i & |s_i| < t, \\ -t & |s_i| > t, \\ \text{undefined} & |s_i| = t. \end{cases} \tag{6.39}$$

Since only weak differentiability is required, we can modify $g(s)$ on sets of measure zero and set $[g(s)]_i = -s_i$ for $|s_i| = t$. Therefore,

$$\|g(s)\|^2 = \sum_{i=1}^{n} \min(|s_i|, t)^2, \tag{6.40}$$

$$\frac{\partial}{\partial s_i} [g(s)]_i = \begin{cases} -1 & |s_i| \leq t, \\ 0 & |s_i| > t, \end{cases} \tag{6.41}$$

and the unbiased estimate of the risk is

$$\operatorname{SURE}(t; s) = n + \sum_{i=1}^{n} \min(|s_i|, t)^2 - |\{1 \leq i \leq n \mid |s_i| \leq t\}|. \tag{6.42}$$

Based on this risk estimate, one then chooses the threshold such that the estimated risk is minimal.

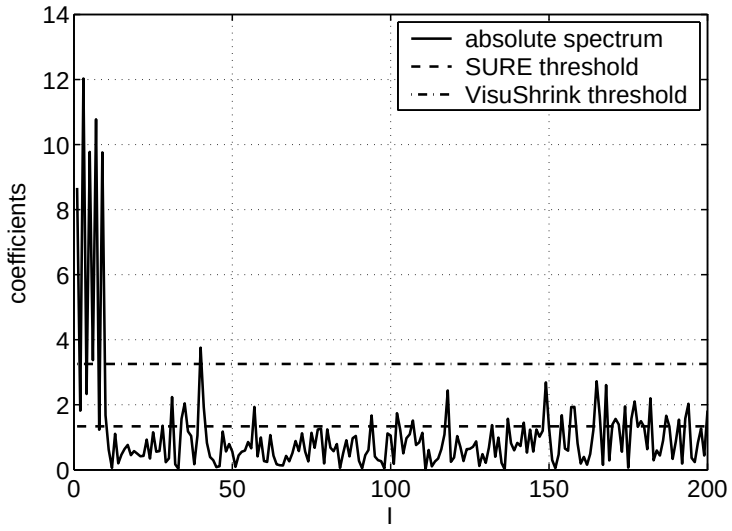


Figure 6.8: Threshold estimates for VisuShrink and SureShrink for the noisy sinc data set. The plot shows the spectrum of the label vector and the estimated thresholds. SureShrink fails to provide good estimates because the interesting coefficients are rather sparse.

Now for the spectrum coefficients of a smooth label vector, we know that the interesting information is contained in the first few leading coefficients, unlike in the wavelet setting, where non-zero coefficients can be found throughout the coefficient set. Therefore, our interest lies more in the cut-off dimension than in identifying individual coefficients which should be recovered in the reconstruction.

The threshold estimation methods can nevertheless be employed for this mean by taking the maximal index which is above the threshold as the cut-off dimension. This will of course lead to problems, if there exist coefficients with a large index which lie accidentally above the threshold.

Furthermore, for the SureShrink method, another problem arises. As discussed by Donoho and Johnstone (1995), the estimates become unreliable in the case where the coefficients are sparse. Therefore, the authors of that paper propose a hybrid method which estimates the sparsity and switches to VisuShrink for sparse solutions. However, in our case, solutions will typically be sparse, and moreover, the sparsity will increase for large sample sizes, because the number of largely expressed coefficients will stay nearly constant while the number of coefficients which contain only noise will increase linearly. To illustrate the poor performance of SureShrink on the data sets arising for the kernel setting, Figure 6.8 plots a typical spectrum for the sinc example with $n = 200$. A significant number of noise points lie above the threshold. On the other hand, the threshold considered by VisuShrink lies on an appropriate level (although there also exists one coefficient above the threshold). We conclude that SureShrink is generally not applicable because the true sequence θ will in general be sparse. VisuShrink on the other hand provides good estimates. However, even for this case, some form of post-processing is advisable to derive a cut-off dimension. Otherwise, the dimension will accidentally be overestimated significantly.

Finally, let us consider the general adoption of threshold shrinkage schemes for regression. As we will discuss in the next chapter, kernel ridge regression works by reconstructing a fit by retaining the leading coefficients of the spectrum and shrinking the remaining indices to zero. The coefficients used for the reconstruction are always a certain number of leading coefficients. In the wavelet setting, these can be any coefficients. Now a significant difference between the wavelet basis and the eigenvector basis is that the wavelet basis functions are localized well. Therefore, one coefficient has only limited effect on the overall fit function. On the other hand, the eigenvectors of a kernel matrix are typically not localized. Therefore, if one erroneously includes a coefficient with large index, this means that the overall fit function is contaminated by an eigenvector of a

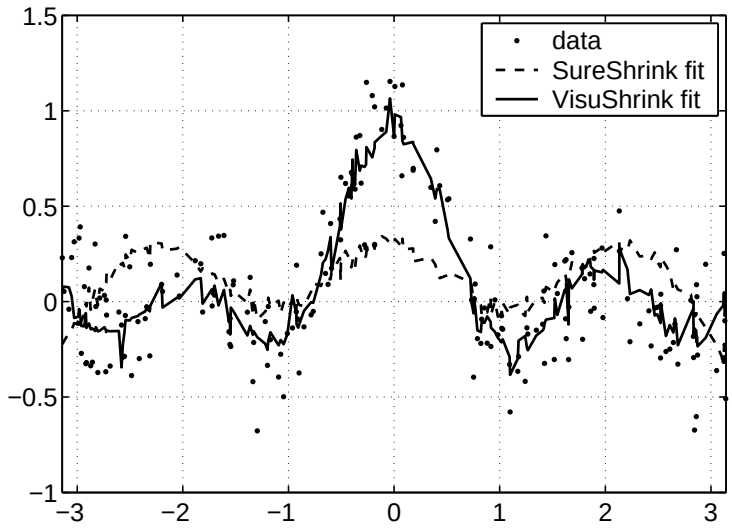


Figure 6.9: Reconstructed fits for the VisuShrink and SureShrink thresholds from Figure 6.8. The fits are inferior through the inclusion of high order coefficients which result in noisy contaminations.

small eigenvalue, which will usually be very irregular. Therefore, the penalty for such an inclusion is much larger than in the wavelet case. In Figure 6.9, the resulting fits are plotted which are obtained from VisuShrink and SureShrink fits. We see that the VisuShrink fit is in principle good, but has unfortunately be contaminated by a coefficient with large index. On the other hand, the SureShrink fit is clearly inferior. Not only is the fit much too irregular, it is also significantly too small.

In summary, although both settings, regression by kernel methods and wavelets, lead to an estimation problem in sequence space, the shrinkage methods from the wavelet approach do not perform well for the kernel based regression.

6.5.4 Experiments

To compare these two methods for estimating the cut-off dimension, we test them on the noisy sinc function example. Recall that X_i is drawn uniformly from $[-\pi, \pi]$. The labels are given as

$$Y_i = \operatorname{sinc}(4X_i) + \sigma_\varepsilon \varepsilon_i, \tag{6.43}$$

$\varepsilon_i \sim \mathcal{N}(0,1)$, such that σ_ε is the standard deviation of the noise. We use the rbf-kernel with scale parameter w, $k(x,y) = \exp(-|x-y|^2/2w)$.

To study the algorithms under various conditions we estimate the cut-off dimension with both algorithms for any combination of parameters from

$$\begin{aligned} n &\in \{100, 200, 500\}, \\ \sigma_\varepsilon &\in \{0, 0.1, 0.2, 0.5\} \\ w &\in \{0.1, 1.0, 2.0, 5.0\}. \end{aligned}$$

Thus, the following conditions are tested: Small versus large sample size, no noise versus large noise, and small versus large kernel widths.

For each combination, we plot histograms of the estimated dimensions over 100 realizations of the data in Figures 6.10 and 6.11. For the remainder of this chapter we will abbreviate the resampling based modality estimate method by RMM and the two component method with TCM.

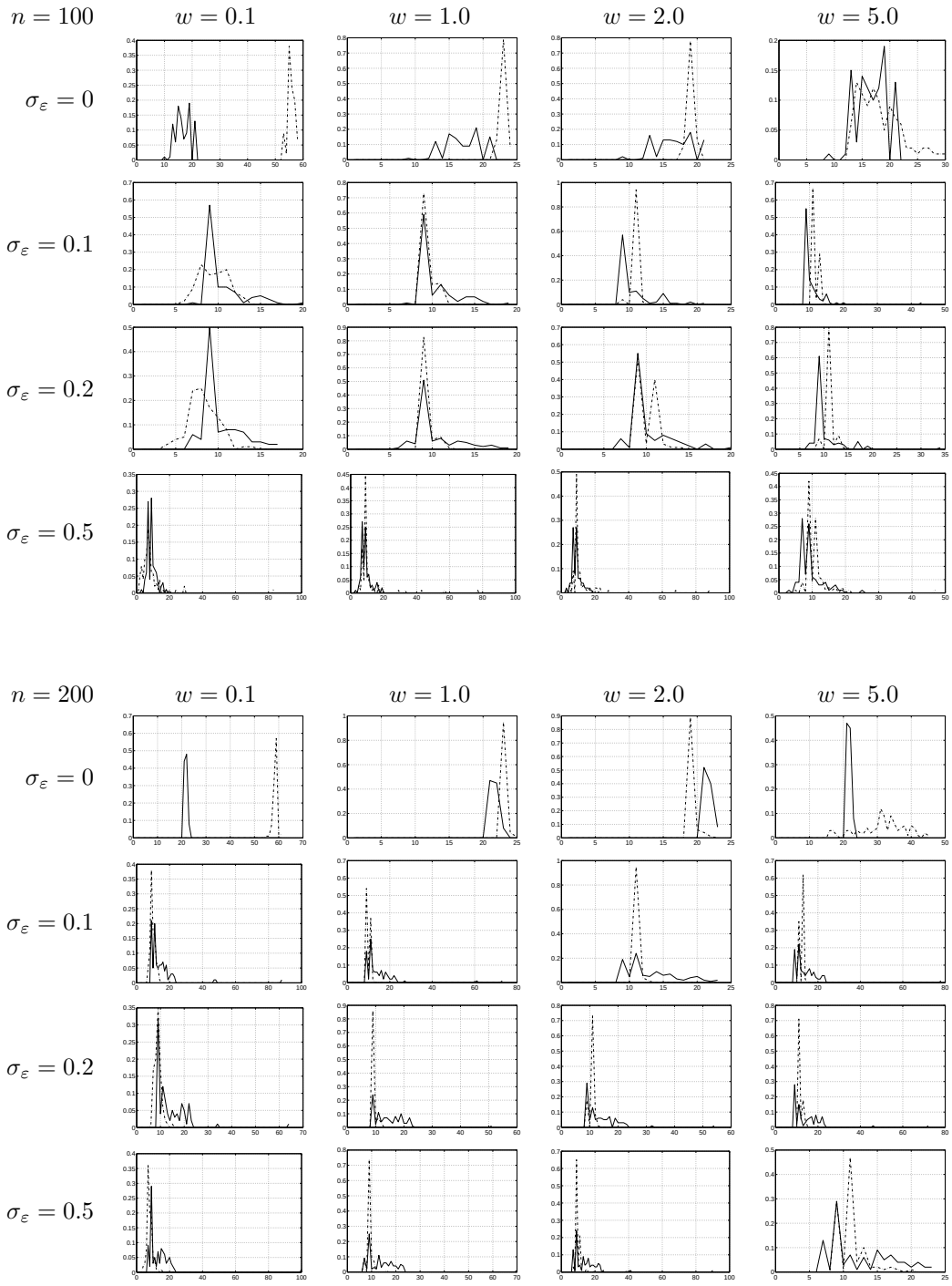


Figure 6.10: Histograms over estimated dimensions for the noisy sinc data set. Solid line: modality estimate with resampling, dashed line: two component model. The two component model estimates are much more stable than those based on the modality estimate.

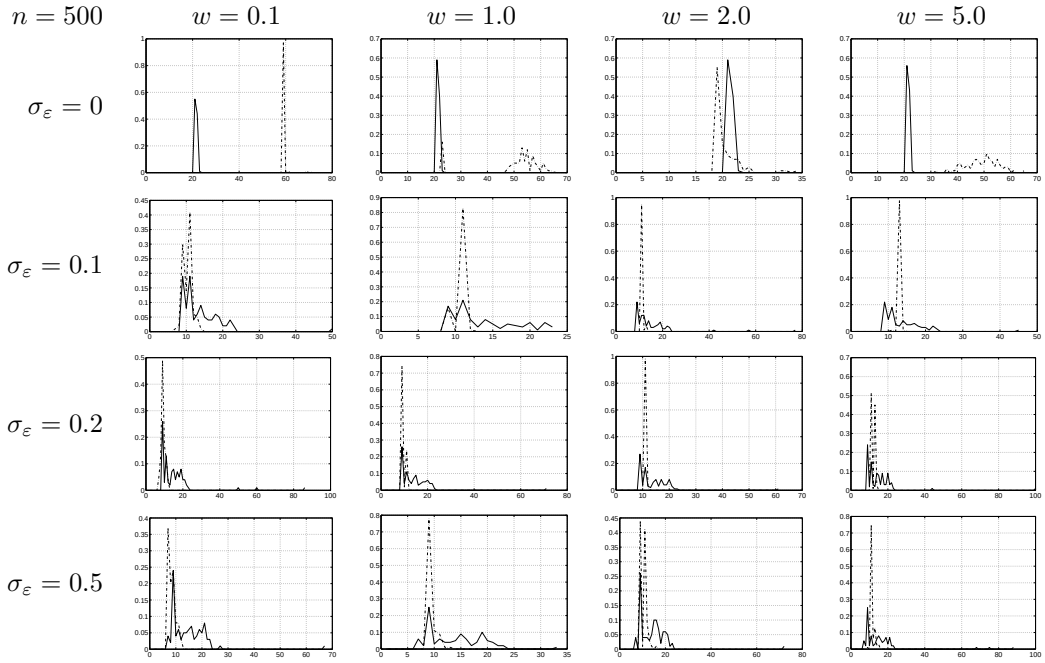


Figure 6.11: (cont'd) Histograms over estimated dimensions for the noisy sinc data set. Solid line: modality estimate with resampling, dashed line: two component model. The two component model estimates are much more stable than those based on the modality estimate.

Over all, the stability of the estimates improves with increasing sample size which is not very surprising. For medium and high noise levels, both methods have a peak at roughly the same cut-off dimension. Note though, that the true cut-off dimension is not at 9 for different kernel width, since the cut-off dimension depends on the spectrum which depends on the eigenbasis of the kernel. This changes with varying kernel widths.

There are two interesting effects. First of all, for $\sigma_\varepsilon = 0$, the estimate of the RMM is roughly the same as for $\sigma_\varepsilon = 0.1$, but the estimates of the TCM change. For small kernel widths, a much higher dimension is estimated around 50–60, while for large kernel width, the estimate becomes very unstable. This behavior is the consequence of many of the higher index spectrum coefficients being very small (around 10^{-10}, see Figure 6.1(c)), such that the TCM is tempted to place one component in that area alone, leading to very high dimensions.

Another effect which is clearly visible is that RMM is in general less stable. For virtually all plots, the estimate of the TCM is much more concentrated. Even for $n = 500$ sample sizes, the RMM occasionally estimates a much larger dimension, even up to 80.

Judging from these experiments, TCM seems to be favored. We will continue the comparison of RMM and TCM after the next section.

6.6 Structure Detection

We propose a further application of the cut-off dimension estimators. Kernel methods are often used with a family of kernel functions which depend on a scale parameter. An example are the radial basis function kernels (rbf-kernels), $k(x, y) = \exp(-|x - y|^2/2w)$, where w is the kernel width. The idea is that the cut-off dimension for a given kernel width w measures the amount of structure present at this scale. Varying the kernel width leads to a structural analysis at different scales. We study how the estimated cut-off dimension d changes with varying kernel width on a data set which contains structure at different scales.

The data set is given by sampling X_i uniformly from $[-\pi, \pi]$, and setting

$$Y_i = \operatorname{sinc}(4X_i) + 0.2\sin(15X_i) + \sigma_\varepsilon \varepsilon_i, \tag{6.44}$$

where ε_i is $\mathcal{N}(0,1)$-distributed. In words, the labels Y_i are formed by superimposing a sinc-function with a high frequency sine wave and additive noise. Figure 6.12 plots a typical realization of the data set for $n = 500$ points.

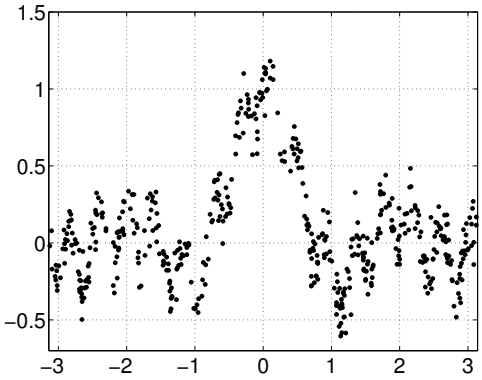


Figure 6.12: A data-set with structure at different scales, $Y_i = \operatorname{sinc}(4X_i) + 0.2\sin(15X_i) + 0.1\varepsilon_i$.

We perform two simulations. For both, we generate a data set according to (6.44) and then estimate the cut-off dimension for 20 kernel widths spaced logarithmically from 10^{-2} to 10. For the first simulation, we keep the sample size fixed at $n = 500$, but vary the noise over

$$\sigma_\varepsilon \in \{0, 0.01, 0.02, \ldots, 0.18, 0.2\}. \tag{6.45}$$

For the second simulation we fix $\sigma_\varepsilon = 0.1$ and let

$$n \in \{100, 150, 200, \ldots, 450, 500\}. \tag{6.46}$$

These computations are carried out over single realizations of the data set, without iterating of several realizations.

Varying noise levels In Figure 6.13, the estimated cut-off dimensions are plotted for both methods. We again see that the RMM is less stable than the TCM resulting in spurious estimates which are much larger than the other estimates. We also see again that the TCM estimate becomes quite large for small kernel widths and no noise.

The most apparent difference between both methods is that the estimates of RMM change smoothly with varying kernel width (apart from the outliers), while the estimates of TCM are more or less constant for small kernel widths and then suddenly change for larger kernel widths. This seems to reflect the idea better that at a certain scale, the finer structure of the high-frequency waves becomes visible.

Next we plot the projections of Y to the space spanned by the first d eigenvectors of the kernel matrix. We have already encountered this projection in Lemma 6.19. Recall that

$$\pi_d Y = \sum_{i=1}^{d} u_i u_i^\top Y. \tag{6.47}$$

Thus, $\pi_d Y$ is the component of Y contained in the space spanned by the first d eigenvectors. Figure 6.14 plots the resulting fits for a small and a large kernel width for selected values of σ_ε. We see that although the cut-off dimension estimated by RMM depends smoothly on the kernel width, at small kernel widths, the sine component is recovered while at large kernel widths, the sinc component is recovered.

Figure 6.15 plots the corresponding fits for the TCM. These plots again nicely recover the structure at both scales in the data.

Varying sample size Figure 6.16 again plots the estimated cut-off dimensions for both methods. For small sample sizes, some patches are missing for RMM. On these data sets, RMM failed to estimate a cut-off dimension, because all coefficients appeared to be unimodal. This shows a severe drawback of RMM especially at small sample sizes. Apart from that, the two plots display the same traits already discussed in the last setting: The TCM cut-off dimension estimates show a clear step, and RMM is generally less stable.

Figures 6.17 and 6.18 plot the projections to the cut-off dimension for some choices of sample sizes. With the exception of RMM for $n = 100$, the plots nicely capture the structure at a large and a small scale.

In summary, using a family of kernel functions depending on a scale parameter, as for example rbf-kernels, one can analyze the structural content of the label vector and obtain the structure at different scale. From the experiments, TCM seems to be more fit for this method due to the following reasons: First of all, TCM is less computationally expensive than RMM. Furthermore, the TCM estimates are generally more stable than the RMM estimates. Finally, the TCM estimates show a behavior similar to a phase transition with varying kernel widths: the estimated cut-off dimension is more or less constant for larger regions of kernel widths, displaying a rapid transition over a relatively short interval. This allows us to detect if there exists structure on different levels. Using RMM, this would be harder to detect because there is no sharp transition.

6.7 Conclusion

We treated a regression setting where the labels are generated by sampling a smooth function and then perturbating the sampled values by adding zero mean noise. We showed that such label vectors represented with respect to the basis of the kernel matrix has a certain structure: From the relative-absolute envelopes on scalar products with eigenvectors from Chapter 4, it follows that the coefficients of the sample vector in the eigenbasis of the kernel matrix decay rapidly and are large only for a finite number of entries, this number being independent of the sample size. On the other hand, the noise component gives rise to evenly distributed coefficients. This means that the signal part can be effectively separated from the noise part. The number of leading coefficients which contains the signal was defined as the cut-off dimension.

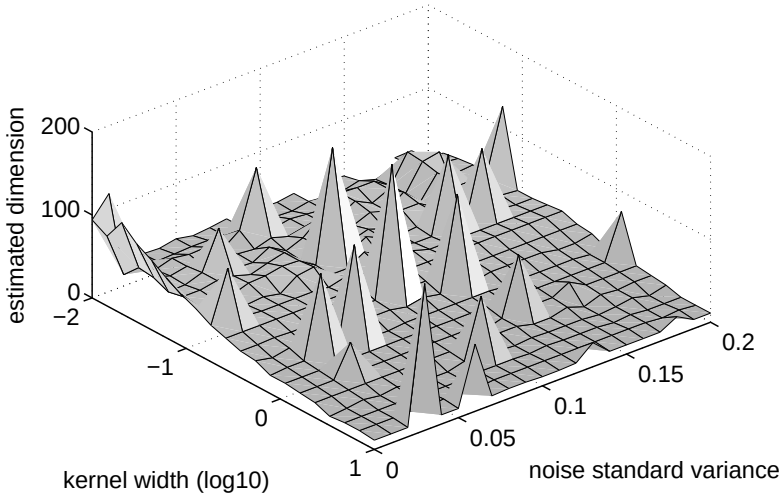
The fact that the information is mostly contained in the first few coefficients of the spectrum has independently already been applied to devising new learning methods. In a paper by Zwald et al. (2005), a support vector machine is trained on the data set after it has been projected to the space spanned by the first few eigenvectors. However, that paper lacks the rigorous justification of this approach by showing that also in the finite sample case, the signal is contained in the leading coefficients of the spectrum.

We have highlighted a conceptual similarity to the framework of wavelet shrinkage, where one also considers the regression problem after a basis transformation, called the sequence space. However, although the situation is very similar, due to substantial differences in the structure of a wavelet basis and the eigenbasis of a kernel matrix, the threshold approaches are not directly applicable to the case of kernel based regression.

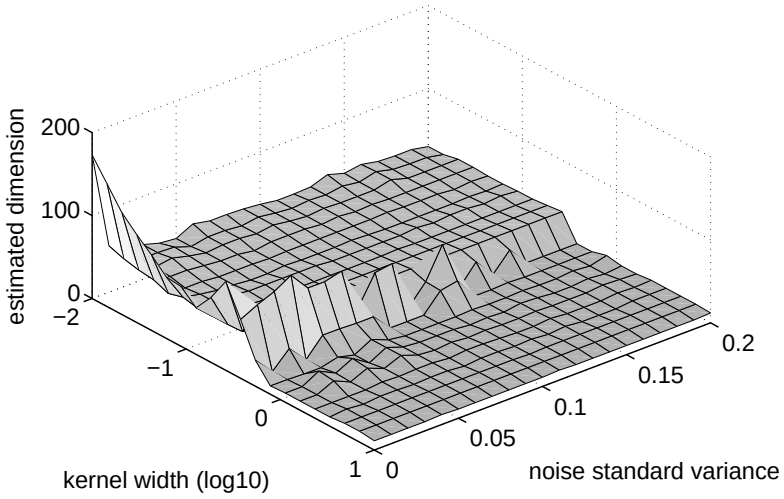
We proposed two procedures for estimating the cut-off dimension, a modality test based on resampling (RMM) and a maximum likelihood approach using a two component model (TCM). Both methods were compared experimentally. Furthermore, we proposed using these cut-off dimension estimators in conjunction with a family of kernel functions with a scale parameter to analyze the structure of the labels at different scales.

Based on these experiments and the simulations on the noisy sinc data set for a number of different settings, we can now undertake a final comparison of RMM and TCM. Overall, TCM was more stable than RMM with the exception of the zero noise case. RMM was even unstable for moderate noise levels and large sample sizes. Moreover, RMM is computationally much more expensive than TCM. For the structure detection application, TCM is also a better choice than RMM because the cut-off dimension of TCM is more or less constant on larger scale regions and

has sharp transitions in between. This makes it easier to detect that there actually exists structure on multiple levels. Thus, the TCM will be used in model selection in Chapter 7.

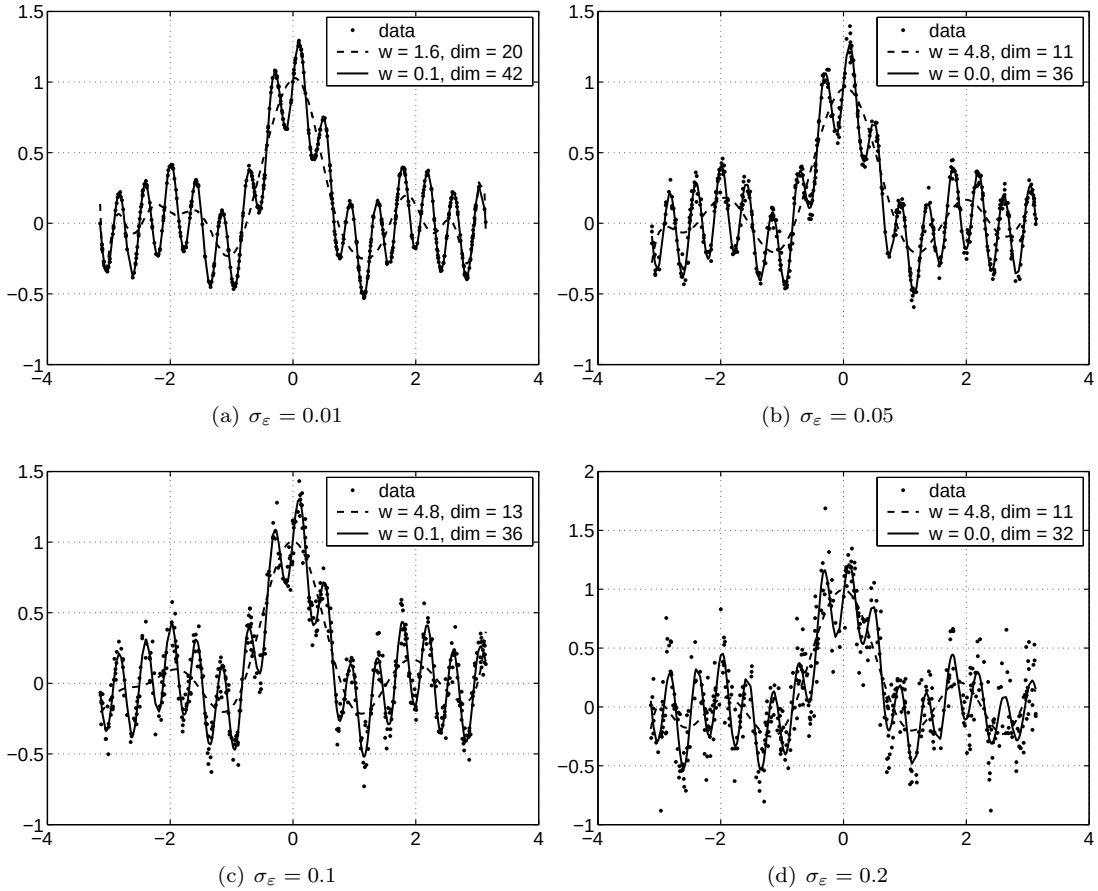


(a) Resampling Based Modality Estimate (RMM)



(b) Two Component Model (TCM)

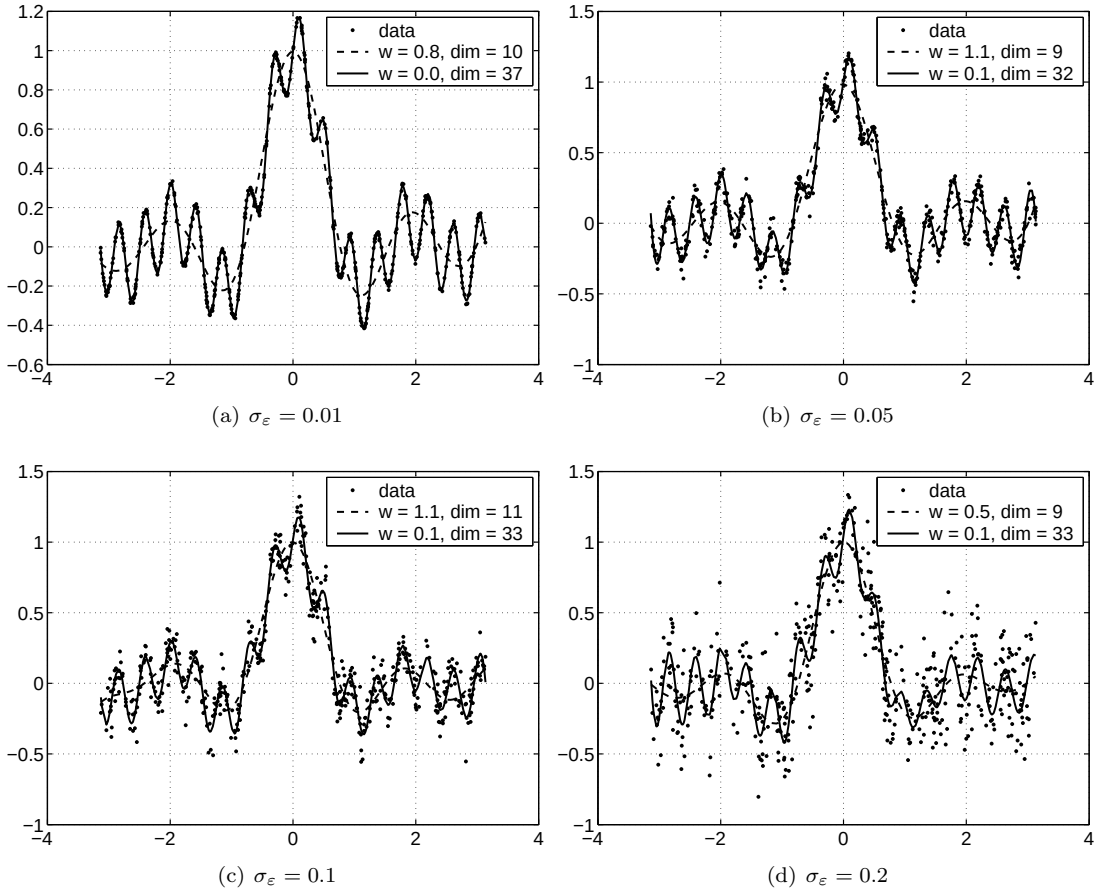
Figure 6.13: Structure detection example: Estimated dimensions for increasing noise variance.



(a) $\sigma_\varepsilon = 0.01$ (b) $\sigma_\varepsilon = 0.05$

(c) $\sigma_\varepsilon = 0.1$ (d) $\sigma_\varepsilon = 0.2$

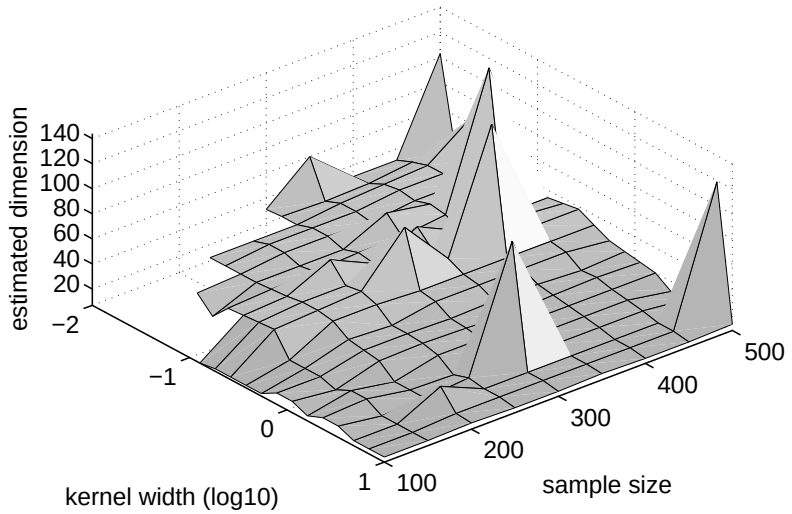
Figure 6.14: Structure detection example: Reconstructions using the estimated dimensions for different noise levels (resampling based modality test).



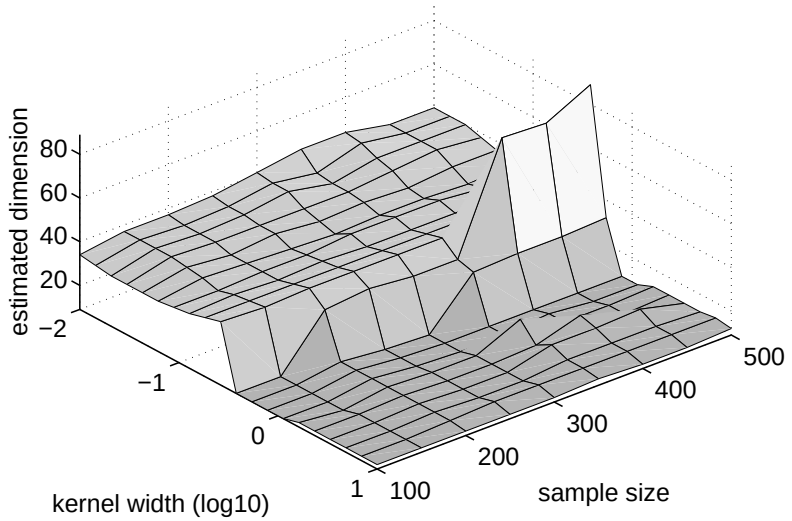
(a) $\sigma_\varepsilon = 0.01$ (b) $\sigma_\varepsilon = 0.05$

(c) $\sigma_\varepsilon = 0.1$ (d) $\sigma_\varepsilon = 0.2$

Figure 6.15: Structure detection example: Reconstructions using the estimated dimensions for different noise levels (two component model).

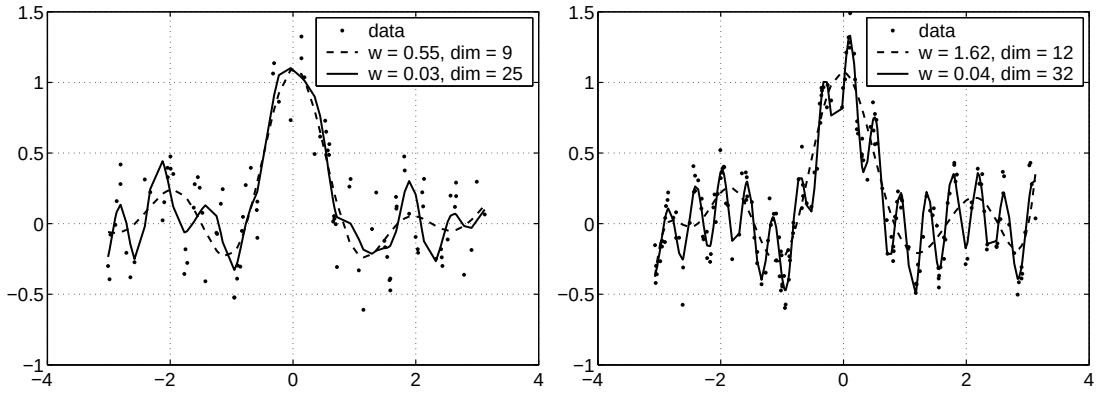


(a) Resampling Based Modality Estimate (RMM)



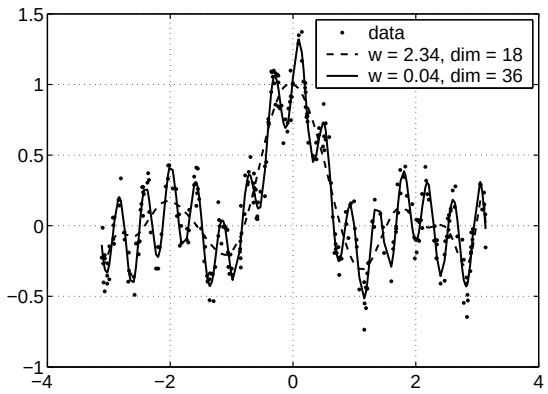
(b) Two Component Model (TCM)

Figure 6.16: Structure detection example: Estimated dimensions for increasing sample size.

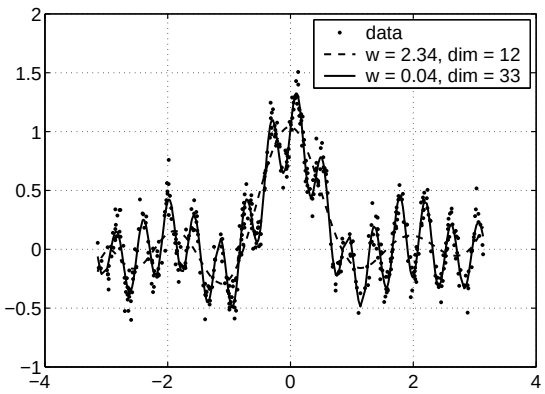


(a) $n = 100$

(b) $n = 200$

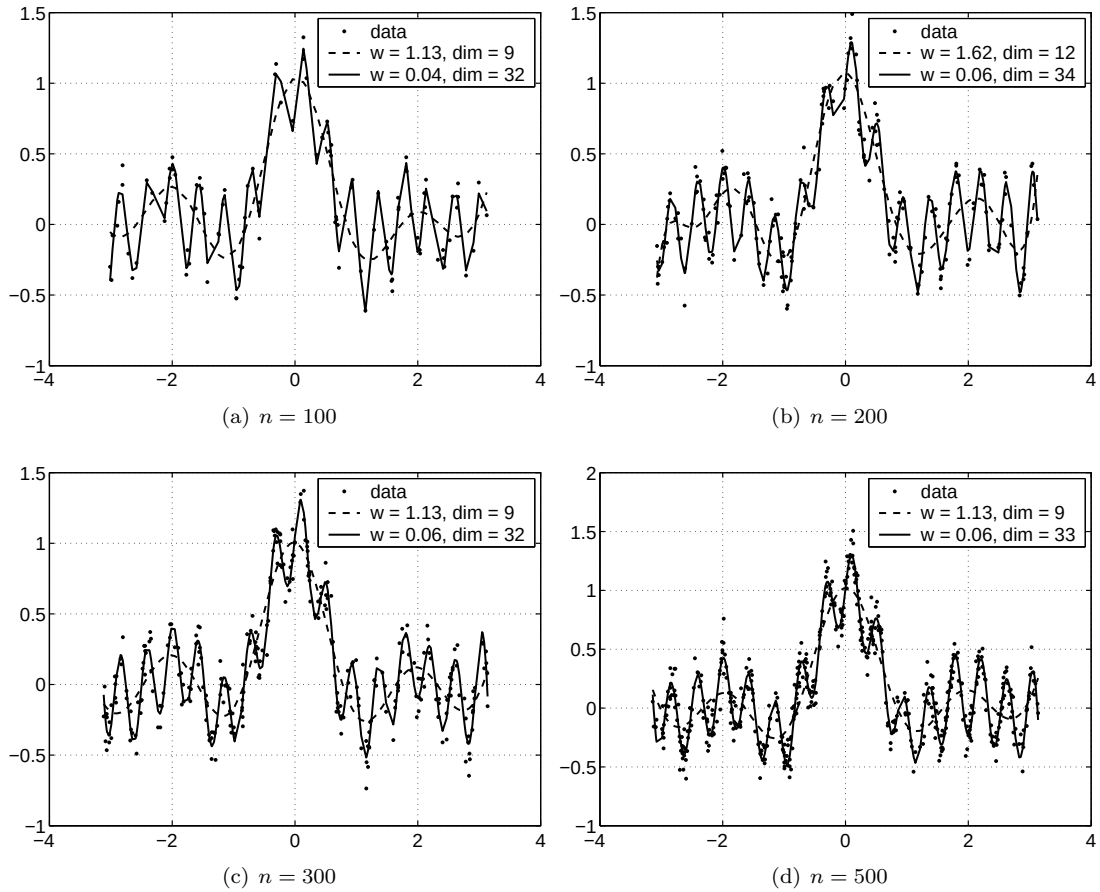


(c) $n = 300$



(d) $n = 500$

Figure 6.17: Structure detection for different sample sizes (resampling based modality test).



(a) $n = 100$ (b) $n = 200$

(c) $n = 300$ (d) $n = 500$

Figure 6.18: Structure detection example: Reconstructions using the estimated dimensions for different sample sizes (two component model)

Chapter 7

Kernel Ridge Regression

Abstract

Kernel ridge regression (KRR) is a standard kernel method for regression. In this section, this algorithm is analyzed based on the results obtained so far. This analysis shows that KRR basically reconstructs the information part of the labels and suppresses the noise. The amount of noise suppression is based on the cut-off dimension. We address the question of model selection using the cut-off dimension estimators from the previous chapter. Experiments show that this approach is in fact able to perform competitively. These observations underline the practical usefulness of the theoretical results.

7.1 Introduction

Kernel methods have proven to be effective and versatile methods for both supervised and unsupervised learning problems. For supervised learning, examples include support vector machines (see for example (Burges, 1998; Müller et al., 2001)) of various kinds, kernel ridge regression (see for example (Cristianini and Shawe-Taylor, 2000)), and a number of additional variants of the procedure in which the fit is minimized and the penalty is computed. Kernel methods also occur in the context of Bayesian inference in the form of Gaussian process regression (see for example (Williams and Rasmussen, 1996; Goldberg et al., 1998)). Common to all these methods is that the computed fit function can be written as

$$\hat{f}(x) = \sum_{i=1}^{n} k(x, X_i)\hat{\alpha}_i + \hat{\alpha}_0, \tag{7.1}$$

where k is the kernel function and $\hat{\alpha}$ the parameter vector where $\hat{\alpha}_0$ may be present or not. Methods differ in the way in which the parameter vector $\hat{\alpha}$ is determined. Among the kernel methods, kernel ridge regression is distinguished by the fact that $\hat{\alpha}$ depends linearly on the label vector Y, which is the vector containing all the labels $Y_1, \ldots, Y_n$. This means that there exists a fit matrix $\mathbf{S}$ such that

$$\hat{\alpha} = \mathbf{S}Y. \tag{7.2}$$

This is not the case for support vector machines and many other methods whose training step involves solving linear programs, quadratic programs or other forms of iterative processes.

That the training step basically consisting in a linear mapping seems to be a great advantage for analyzing how the learning algorithm works. Moreover, the fit matrix is symmetric and closely linked to the kernel matrix, whose structure has already been extensively analyzed in this thesis. In the first part of this chapter, we will analyze KRR using these results. We will show that kernel ridge regression basically learns a fit function by first performing a basis transformation into the eigenbasis of the kernel matrix. In this representation, the information content is contained in the first few leading coefficients, while the noise is spread evenly over all coefficients. KRR then

retains only a number of leading coefficients shrinking the remaining coefficients effectively to zero. Thus, the noise is removed and the target function is learned.

Kernel learning methods for supervised learning invariably come with certain free parameters which have to be adjusted during the training step. One of these parameters is the choice of kernel function, or, if a family of kernel functions is employed, the choice of the parameter of the family. For example, when using the rbf-kernel, the kernel width has to be adjusted.

For kernel ridge regression (KRR) used in conjunction with the rbf-kernel, two parameters exist: the kernel width w and the regularization parameter τ. The standards methods for estimating good choices for these parameters fall in one of the following three categories:

The parameters are estimated by some form of hold-out testing, also known as *k-fold cross validation*. This means that one estimates the true generalization error by iteratively removing a subset of the training set, training on the remaining set and calculating the loss on the hold out test set. In the most extreme case, only one point is removed from the training set. This procedure is known as *leave-one-out cross-validation*. The parameter set which leads to the smallest test error is then selected to perform the training on the whole set. Note that this procedure implies doing a full grid search of all parameters because the test error is usually not convex and several local minima exist. Therefore, if more than one parameter is involved, hold-out testing can become quite time consuming. Fortunately, for KRR, once the spectral decomposition of the kernel matrix has been computed, the leave-one-out test errors for different values of τ can be computed in $O(n^2)$ for each τ, which is much less than doing a full training step (typically in $O(n^3)$).

The second category is given by methods which estimate the generalization error by adding penalties to the training error to account for the optimism of the training error. These procedures are often based on some approximation which depends on assumptions which need not hold in general such that these models are not very robust for certain data configurations.

Finally, in the framework of Gaussian processes, the free parameters are model parameters, which can be inferred from the data as well. Below, we will discuss the method of performing this estimation by maximizing the marginal likelihood on the training example. Then, the parameters which maximize the likelihood are chosen for the final training step. This approach might be problematic if the modelling assumption is not met, or if only a subset of the parameters should be adjusted.

In the second part of this chapter, we wish to explore whether the structural insights developed so far can be used to estimate the parameters without performing neither hold-out testing, nor using penalty terms, nor using maximum likelihood approaches. If this is possible, it shows that our theoretical results actually translate to effective procedures in practice, and that the theoretical results really describe the actual behavior of the algorithm well.

On the downside, we have to be honest enough to admit that there is no real need for new model selection algorithms for kernel ridge regression since the existing methods are efficient and work very well in practice. However, our method has the added benefit of providing additional structural information about the data set as, for example, the number of effective dimensions and the variance of the noise. This analysis gives an indication of the hardness of the learning problem which contains more information than the pure test error.

This chapter is structured as follows. Section 7.2 reviews the main results of this chapter. Section 7.3 discusses KRR based on the results on the spectral properties of the kernel matrix. Existing approaches for model selection for KRR are reviewed in Section 7.4. The spectrum method is introduced in Section 7.5. Experimental results are presented in Section 7.6 for regression and in Section 7.7 for classification. Section 7.8 concludes this chapter.

7.2 Summary of Main Results

In the present chapter, we will discuss the kernel ridge regression (KRR) algorithm in view of the results from previous chapters. We rewrite KRR in terms of the eigenvalues and eigenvectors of the kernel matrix. Using the results from Chapters 3 and 4, we can provide accurate descriptions

of how kernel ridge regression works. KRR basically consists of three steps: First the coefficients of the label vector with respect to the eigenbasis of the kernel matrix are computed. This quantity has been called the *spectrum* in Chapter 6. As discussed in that chapter, the label vector has a peculiar structure in this representation: The sample vector of the target function is contained in the first few leading coefficients of the spectrum while the noise is distributed evenly over all coefficients. The second steps weights the spectrum coefficients with the factor $l_i/(l_i+\tau)$. We can infer based on the results on the eigenvalues of the kernel matrix from Chapter 3 that these factors are close to 1 for a number of leading eigenvalues, after which the factors quickly shrink to zero. In conjunction with the knowledge on the structure of the spectrum of the label vector, we know that if the regularization parameter τ is properly adjusted, all but the first few coefficients which contain the sample vector of the target function are set to zero such that the noise is effectively removed. Then, the resulting fit is reconstructed. In summary, we see that KRR works because it transforms the label vector into a representation where the noise can easily be removed.

Based on this analysis, it seems reasonable to adjust the regularization parameter such that the number of recovered dimensions matches the cut-off dimension of the label vector. In the second part of this chapter, we propose a method for estimating the regularization parameter τ, called the *spectrum method*. The spectrum method uses the estimated cut-off dimension from Chapter 6 to adjust the regularization constant τ.

In order to adjust the kernel widths, the generalization error is estimated using the leave-one-out cross-validation error given the regularization constant returned by the spectrum method. Then, the kernel width is chosen which results in the smallest cross-validation error.

We compare the spectrum methods against a maximum marginal likelihood (GPML) approach from Gaussian processes and leave-one-out cross-validation. We perform an experimental comparison on the noisy sinc toy data set. For fixed kernel widths, it turns out that the spectrum method is more robust against the choice of the wrong kernel method than the maximum marginal likelihood approach. It seems that GPML tries to compensate for the choice of a too large kernel with the regularization constant which amounts to an estimate of the noise variance in the Gaussian process framework.

We then compare the algorithms on the `bank` and `kin` benchmark data sets from the DELVE repository. Here, all three algorithms perform very similarly, showing that the spectrum method is on par with state-of-the-art methods.

Although the spectrum method has been derived in a regression context, we wish to test how well the method performs in a classification context. We tested the algorithms on the benchmark data sets from Rätsch et al. (2001). Again, the spectrum method is on par with the existing methods and can even produce slightly better results on some data set than the expensively hand-tuned support vector machine from the original publication.

7.3 Kernel Ridge Regression

In the first part of this chapter, we introduce kernel ridge regression and discuss it based on the results obtained in previous chapters.

7.3.1 The Kernel Ridge Regression Algorithm

Kernel ridge regression (KRR) is a method for computing a smooth fit function from a set of noisy examples. The input to the algorithm is a training set $(X_1, Y_1), \ldots, (X_n, Y_n)$, where the X_i lie in some space $\mathcal{X}$, and the Y_i are in $\mathbb{R}$. As additional parameters, KRR needs a Mercer kernel function $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and a regularization parameter $\tau > 0$. The output is a fit parameter vector $\hat{\alpha}$ which can be used in conjunction with the training points $X_1, \ldots, X_n$ to compute the fit function $\hat{f}$ by

$$\hat{f}(x) = \sum_{i=1}^{n} k(x, X_i)\hat{\alpha}_i. \tag{7.3}$$

Kernel Ridge Regression

Input:	$X_1, \ldots, X_n \in \mathbb{R}^d$, $Y = (Y_1, \ldots, Y_n)^\top \in \mathbb{R}^n$.
Parameters:	a Mercer kernel function $k\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ regularization parameter $\tau \in \mathbb{R}_{>0}$.
Output:	parameter vector $\hat{\alpha}$, function $\hat{f}\colon \mathbb{R}^d \to \mathbb{R}$.

1 Compute kernel matrix $\mathbf{K}$ with $[\mathbf{K}]_{ij} = k(X_i, X_j)$.
2 Solve $(\mathbf{K} + \tau\mathbf{I})\alpha = Y$, giving solution $\hat{\alpha}$
3 Return $\hat{\alpha}$ and
$\hat{f}(x) = \sum_{i=1}^{n} k(x, X_i)\hat{\alpha}_i$.

Figure 7.1: The kernel ridge regression algorithm.

The fit is computed by

$$\hat{\alpha} = (\mathbf{K} + \tau\mathbf{I})^{-1} Y, \tag{7.4}$$

where $Y = (Y_1, \ldots, Y_n)^\top$, and $\mathbf{K}$ is the kernel matrix. The resulting algorithm is summarized in Figure 7.1.

The main computational costs are clearly induced by inverting the matrix $\mathbf{K} + \tau\mathbf{I}$. This matrix is symmetric and in general neither sparse nor conditioned well. One of the standard methods has thus to be used leading to a time complexity of $O(n^3)$ (Golub and van Loan, 1996). A less demanding step is the computation of the kernel matrix itself. For example, for the ubiquitous rbf-kernel fast algorithms for the computation of the exponential function can be employed (Ahrendt, 1999).

The regularization parameter will often by quite small (up to 10^{-8}), such that the resulting $\hat{\alpha}$ might not be very stable. On the other hand, the final fit $\hat{f}$ is obtained by mapping $\hat{\alpha}$ to $\mathbf{K}\hat{\alpha}$, which stabilizes the components of $\hat{\alpha}$ which are likely to be unstable.

It is possible to use a conjugate gradient method to solve $(\mathbf{K} + \tau\mathbf{I})\alpha = Y$, since $\mathbf{K}$ is positive definite and symmetric. Unfortunately, since $\mathbf{K}+\tau\mathbf{I}$ might not be conditioned well, a conjugate gradient method may take as much time as one of the standard methods. If one is interested only in the fit on y, one has to solve $(\mathbf{K} + \tau\mathbf{I})\hat{Y} = \mathbf{K}Y$. Conjugate gradient methods converge much faster on this problem.

KRR can be motivated in different contexts, either as ridge regression in feature space (Cristianini and Shawe-Taylor, 2000), via Gaussian processes (Williams and Rasmussen, 1996), or by Regularization Networks (Girosi et al., 1995). We will propose an independent explanation of KRR based on the results obtained so far in this thesis.

7.3.2 Spectral Decomposition of Kernel Ridge Regression

First of all, since the results we have obtained so far are mainly in connection with the eigenvalues and eigenvectors of the kernel matrix, we will derive a representation of the fit function $\hat{f}$ computed by kernel ridge regression in terms of the spectral decomposition of the kernel matrix.

We will first consider the *in-sample* fit $\hat{Y} = (\hat{f}(X_1), \ldots, \hat{f}(X_n))^\top$, which defines the training error $\|\hat{Y} - Y\|^2/n$. Compute that

$$[\hat{Y}]_i = \hat{f}(X_i) = \sum_{j=1}^{n} k(X_i, X_j)\hat{\alpha}_j = [\mathbf{K}\hat{\alpha}]_i = [\mathbf{K}(\mathbf{K} + \tau\mathbf{I})^{-1} Y]_i. \tag{7.5}$$

Now using the spectral decomposition $\mathbf{K} = \mathbf{U}\mathbf{L}\mathbf{U}^\top$ of the kernel matrix, we can rewrite the in-

sample fit as follows:

$$\begin{aligned}\hat{Y} = \mathbf{K}(\mathbf{K}+\tau\mathbf{I})^{-1} &= \mathbf{U}\mathbf{L}\mathbf{U}^\top(\mathbf{U}\mathbf{L}\mathbf{U}^\top+\tau\mathbf{I})^{-1}Y \\ &= \mathbf{U}\mathbf{L}\mathbf{U}^\top(\mathbf{U}(\mathbf{L}+\tau\mathbf{I})\mathbf{U}^\top)^{-1}Y \\ &= \mathbf{U}\mathbf{L}\mathbf{U}^\top\mathbf{U}(\mathbf{L}+\tau\mathbf{I})^{-1}\mathbf{U}^\top Y \\ &= \mathbf{U}\mathbf{L}(\mathbf{L}+\tau\mathbf{I})^{-1}\mathbf{U}^\top Y \\ &= \sum_{i=1}^{n} u_i \frac{l_i}{l_i+\tau} u_i^\top Y.\end{aligned} \tag{7.6}$$

We see that in order to compute the in-sample fit, the coefficients of Y with respect to the basis of eigenvectors of $\mathbf{K}$ are scaled by $l_i/(l_i+\tau)$. We will discuss this fact in greater detail later on.

For now, we are interested in the question if we can write the whole fit function $\hat{f}$ in a similar fashion, ideally only by extending the first term u_i to a continuous function. This can be achieved as follows. Note that since $l_i u_i = \mathbf{K}u_i$, $u_i = \mathbf{K}u_i/l_i$. We thus define the continuated version of the eigenvector u_i as

$$u_i^c(x) = \frac{1}{l_i}\sum_{j=1}^{n} k(x, X_i)[u_i]_j. \tag{7.7}$$

Note that $u_i^c(X_j) = [u_i]_j$, such that (7.7) can be considered as an interpolation of the points $(X_j, [u_i](X_j))_{j=1}^n$.

Now define the row-vector $k_x = (k(x, X_1), \ldots, k(x, X_n))$. With that,

$$\hat{f}(x) = \sum_{j=1}^{n} k(x, X_j)\hat{\alpha}_j = k_x\hat{\alpha}, \quad \text{and} \quad u_i^c(x) = \frac{1}{l_i}k_x u_i. \tag{7.8}$$

We can write the fit function as follows

$$\hat{f}(x) = k_x\hat{\alpha} \overset{(*)}{=} k_x \sum_{i=1}^{n} u_i u_i^\top\hat{\alpha} = \sum_{i=1}^{n} l_i u_i^c(x) u_i^\top\hat{\alpha} = \sum_{i=1}^{n} u_i^c \frac{l_i}{l_i+\tau} u_i^\top Y. \tag{7.9}$$

At $(*)$, we use the fact that $\{u_1, \ldots, u_n\}$ forms a orthonormal basis of $\mathbb{R}^n$, such that $\sum_{i=1}^n u_i u_i^\top\hat{\alpha} = \hat{\alpha}$. Furthermore, the last step follows from the fact that $\hat{\alpha} = \sum_{i=1}^n u_i(l_i+\tau)^{-1}u_i^\top Y$, which follows from (7.6).

We summarize these computations.

Result 7.10 (Spectral decomposition of the fit) *The fit in kernel ridge regression is*

$$\hat{f}(x) = \sum_{i=1}^{n} u_i^c(x)\frac{l_i}{l_i+\tau}u_i^\top Y, \tag{7.11}$$

where l_i are the eigenvalues of $\mathbf{K}$, u_i the eigenvectors of $\mathbf{K}$, and u_i^c continuous extrapolations of the eigenvectors given by formula (7.7).

Since $u_i^c(X_j) = [u_i]_j$, the fit $\hat{y}$ on the data points $x_1, \ldots, x_n$ is given as

$$\hat{Y} = \sum_{i=1}^{n} u_i \frac{l_i}{l_i+\tau} u_i^\top Y. \tag{7.12}$$

We will now take a closer look at these formulas. So far, the derivation was purely algebraic and it is yet unclear in what sense *learning* is accomplished. Let us take a closer look at the steps involved in computing the fit $\hat{f}$. The computation of $\hat{f}$ can be divided into three parts:

1. Scalar products between u_i and Y are computed. The resulting vector is

$$s = \left(u_1^\top Y, \ldots, u_n^\top Y\right)^\top = \mathbf{U}^\top Y. \tag{7.13}$$

2. The coefficients s_i are weighted by the factor $\frac{l_i}{l_i+\tau}$, which results in the vector

$$s' = \left(\frac{l_1}{l_1+\tau} s_1, \ldots, \frac{l_n}{l_n+\tau} s_n \right)^\top. \tag{7.14}$$

3. The fit $\hat{y}$ is constructed by forming

$$\hat{f} = \sum_{i=1}^{n} u_i^c s_i'. \tag{7.15}$$

Fortunately, we are in a position to already have extensive theoretical results for each of these steps available such that we can understand what the steps amount to.

7.3.3 An Analysis of Kernel Ridge Regression

We will assume that we are in a regression setting as introduced in Chapter 6. Recall that this means that the X_i are i.i.d. samples from some probability distribution μ. The Y_i are defined as

$$Y_i = f(X_i) + \varepsilon_i, \tag{7.16}$$

where $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)$ is distributed as $\mathcal{N}(0, \sigma_\varepsilon^2 \mathbf{I}_n)$. The function f is the target function. We have assumed that this function is smooth in the sense that it is in the range of T_k, the integral operator associated with k defined in (2.18), such that there exists a ℓ^2 sequence $\alpha = (\alpha_\ell)$ with

$$f = \sum_{\ell=1}^{\infty} \alpha_\ell \lambda_\ell \psi_\ell. \tag{7.17}$$

Smoothness of f then follows from Lemma 2.33 and the fact that α has finite norm. The vector $f(\mathbf{X}) = \big(f(X_1), \ldots, f(X_n)\big)^\top$ will be called the *sample vector* of f, while ε will be called the noise vector.

The first step computes the *spectrum* of Y (see Chapter 6). We know that the resulting spectrum vector has a peculiar form: The sample vector $f(\mathbf{X})$ is contained in a few leading coefficients of the spectrum vector and sticks out of the noise, which has an evenly distributed spectrum. Thus, by first computing the scalar products with the eigenvectors of the kernel matrix, the label vector Y is transformed into a representation where the interesting part of the label vector, the sample vector $f(\mathbf{X})$ can be separated easily from the noise part ε. Let us accompany this discussion with plots from an actual example, of fitting the noisy sinc function. In Figure 7.2(a), the data set is plotted. The spectrum for $F = f(\mathbf{X})$ and ε is plotted in Figure 7.2(b). We see how the two components have the different structures as explained.

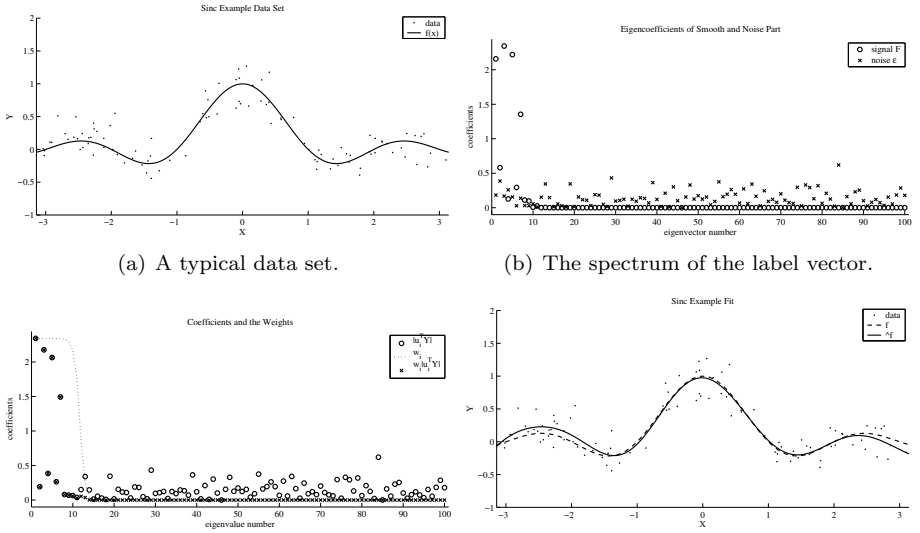
While the first step can be considered a preprocessing step to transform the label vector Y into a representation which makes it more amenable to analysis, the weighting step is where the information is actually processed. The result is an altered version of the spectrum vector which is then used in the final reconstruction step to generate the fit. The goal of kernel ridge regression is of course to preserve the information of the sample vector $f(\mathbf{X})$ of the target function and remove as much noise ε as possible.

The entries of the spectrum are weighted by the factors $w_i := l_i/(l_i+\tau)$. Let us consider this function

$$h(l) = \frac{l}{l+\tau} = \frac{1}{1+\frac{\tau}{l}}. \tag{7.18}$$

We see that $h(l)$ is strictly decreasing as l decreases, and that

$$\lim_{l\to\infty} h(l) = 1, \qquad \lim_{l\to 0} h(l) = 0. \tag{7.19}$$



(a) A typical data set.

(b) The spectrum of the label vector.

(c) The coefficients after the weighting step.

(d) The resulting fit

Figure 7.2: The noisy sinc example.

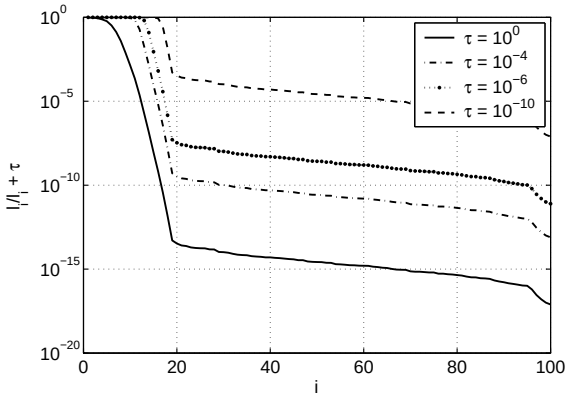


Figure 7.3: Weights for different choices of τ.

The change from 1 to 0 happens for $l < \tau$ and is at least as fast as the decay of the eigenvalues because

$$\frac{l}{l+\tau} \leq \frac{l}{\tau} \wedge 1. \tag{7.20}$$

(Here, "$\wedge$" denotes the minimum.) Now considering the eigenvalues l_i of the kernel matrix, by the results of Chapter 3, we know that l_i will decay as quickly as the eigenvalues λ_i of T_k until they stagnate at around the finite precision ε of the underlying floating point architecture. More specifically, the eigenvalues l_i converge to λ_i with a relative-absolute bound (see Theorem 3.71)

$$|l_i - \lambda_i| \leq \lambda_i C(r,n) + E(r,n), \tag{7.21}$$

where $C(r,n) \to 0$ and r can be chosen such that $E(r,n)$ is as small as the finite precision of the underlying floating point architecture. Thus, the weights will decay as quickly as λ_i and also reach a plateau at ε/τ. In summary, we can expect that the factors are close to one for a finite number of indices and then quickly decay to practically 0. Figure 7.3 plots a typical example for different values of τ. We see how the weights nicely drop as soon as $\lambda_i < \tau$. In Figure 7.2(c), the spectrum is plotted after the weighting step. The part of the spectrum containing $f(\mathbf{X})$ has

been kept invariant, while the remaining spectrum which consists only noise has been shrunk to practically zero.

Finally, in the last step, the extrapolated eigenvectors provide good extrapolations as the u_c are bounded in complexity, and we obtain a fit which is close to the true function (see Figure 7.2(d)). The remaining noise is due to the noise still present in the first few coefficients of the spectrum and due to extrapolation errors.

This kind of interpretation which we have laid out is not entirely new. In fact, in connection with smoothing splines very similar observations can be made (see for example Hastie et al. (2001)). However, based on the results on the spectral properties of the kernel matrix, we can support these observations with rigorous results. In particular, we obtain a theoretical guarantee that the l_i decay quickly and that the spectrum of $f(\mathbf{X})$ has the described properties. Otherwise, for example, it becomes very difficult to guarantee that the eigenvalues of $\mathbf{K}$ actually decay quickly, something which is supremely important for KRR to work.

The open question is the choice of the regularization parameter. Based on the discussion in this Section, it seems advisable to select τ based on the cut-off dimension introduced in Chapter 6. In the remainder of this chapter, we will explore this approach.

7.4 Some Model Selection Approaches for Kernel Ridge Regression

In the second part of this chapter, we will explore whether the cut-off dimension estimators of Chapter 6 can be used to perform effective model selection. We start by briefly review existing state-of-the-art approaches to model selection for kernel ridge regression

7.4.1 Cross Validation

Cross-validation (Cover, 1969) is a general term for any procedure which estimates the generalization error of an algorithm by splitting the training set into two sets, training on one set and testing on the other set. Usually, this step is performed for a number of resamples, leading to an estimate with a certain error margin attached, allowing not only to compare test errors but also to see if they are significantly different or not.

One extreme form of cross-validation is *leave-one-out cross-validation* (LOOCV), where the data set of size n is split $n-1$ times, always removing only a single point, training on the remaining data set and then testing on the one test set. Fortunately, for kernel ridge regression, the leave-one-out error can be computed in closed form, such that it does not involve $n-1$ full training steps.

Let $\mathbf{S}$ be the fit matrix:

$$\mathbf{S} = \mathbf{K}(\mathbf{K} + \tau\mathbf{I})^{-1}. \tag{7.22}$$

The leave-one-out cross-validation error is then (see Wahba (1990))

$$\mathsf{err}_{\mathsf{CV}} = \frac{1}{n} \left\| (\mathrm{diag}(\mathbf{I} - \mathbf{S}))^{-1}(\mathbf{I} - \mathbf{S})Y \right\|^2, \tag{7.23}$$

where $\mathrm{diag}\,\mathbf{A}$ is the matrix $\mathbf{A}$ with off-diagonal elements set to zero. Of course, in order to use LOOCV for model selection, one computes the leave-one-out error for a number of parameter choices and selects the parameter which minimizes the error. Given the spectral decomposition of $\mathbf{K}$, the leave-one-out cross-validation errors for different values of τ can be computed in $O(n^2)$: Let $\mathbf{K} = \mathbf{U}\mathbf{L}\mathbf{U}^\top$. Then,

$$(\mathbf{I} - \mathbf{S})Y = Y - \mathbf{K}(\mathbf{K} + \tau\mathbf{I})^{-1}Y = Y - \mathbf{U}\mathbf{L}(\mathbf{L} + \tau\mathbf{I})^{-1}\mathbf{U}^\top Y. \tag{7.24}$$

The vector $\mathbf{U}^\top Y$ can be computed beforehand. Since $\mathbf{L}(\mathbf{L} + \tau\mathbf{I})^{-1}$ is diagonal, we can compute multiplication with a vector in $O(n)$. Finally, we have to multiply by $\mathbf{U}$ in $O(n^2)$.

The diagonal elements of $\mathbf{I} - \mathbf{S}$ can also be computed in $O(n^2)$. Let $\mathbf{Z} = \mathbf{L}(\mathbf{L} + \tau\mathbf{I})^{-1}$. Then,

$$\begin{aligned}[\mathbf{I} - \mathbf{S}]_{ii} &= 1 - [\mathbf{U}\mathbf{Z}\mathbf{U}^\top]_{ii} \\ &= 1 - \sum_{j,k} [\mathbf{U}]_{ij}[\mathbf{Z}]_{jk}[\mathbf{U}]_{ik} = 1 - \sum_j [\mathbf{U}]_{ij}[\mathbf{Z}]_{jj}[\mathbf{U}]_{ij} = 1 - \sum_j [\mathbf{U}]_{ij}^2[\mathbf{Z}]_{jj},\end{aligned} \tag{7.25}$$

which is a matrix-vector multiplication between the matrix $\mathbf{U}$ with each element squared and the diagonal of $\mathbf{Z}$.

7.4.2 Marginal Likelihood Maximization

The Gaussian process approach is a very successful Bayesian method for regression. It places a Gaussian prior over the space of functions which permits to perform distributional predictions for new data points, that means, given a fixed training example, the prediction at a new point x is not only a single value but a normal distribution. Interestingly enough, the mean of this distribution coincides with the prediction computed by kernel ridge regression. (There exist a large number of publications on Gaussian processes. As a starting point see the papers by Williams and Rasmussen (1996); Goldberg et al. (1998))

The Gaussian prior is a random process on $\mathbb{R}^d$, which is specified by a consistent family of finite dimensional distributions, as customary in the context of random processes with uncountable index spaces. Thus for a finite set of vectors $X_1, \ldots, X_n$ and labels $Y_1, \ldots, Y_n$, the distribution of $(Y_1, \ldots, Y_n)^\top$ is assumed to be normally distributed with mean 0 and some covariance matrix C. One possible choice for C is to use some Mercer kernel function k, plus a term for the independent noise added to each Y_i,

$$\mathbf{Cov}(Y_i, Y_j) = k(X_i, X_j) + \delta_{ij}\sigma_\varepsilon^2. \tag{7.26}$$

This assumes a certain amount of coupling between Y_i and Y_j based on the distance, which translates to a certain amount of smoothness of the inferred functions. Therefore, $Y = (Y_1, \ldots, Y_n)^\top$ has the probability density

$$p(Y) = (2\pi)^{-\frac{1}{2}n}(\det(\mathbf{K} + \sigma_\varepsilon^2\mathbf{I}))^{-\frac{1}{2}} \exp\left(-\frac{1}{2}Y^\top(\mathbf{K} + \sigma_\varepsilon^2\mathbf{I})^{-1}Y\right), \tag{7.27}$$

where $[\mathbf{K}]_{ij} = k(X_i, Y_j)$.

Now the maximum likelihood approach to estimating these parameters consists in maximizing (7.27) for a given data set with respect to any parameters on which k and σ_ε^2 depend. For example, if one uses rbf-kernels, the width has to be adjusted to reflect the covariance structure of the data. This approach will be called GPML.

In order to maximize the likelihood one usually resorts to maximizing the log-likelihood with respect to the kernel and σ_ε^2 which can be easily computed from (7.27) as

$$\text{loglik}(Y|\mathbf{K}, \sigma_\varepsilon^2) = -\frac{1}{2}\log\det(\mathbf{K} + \sigma_\varepsilon^2\mathbf{I}) - \frac{1}{2}Y^\top(\mathbf{K} + \sigma_\varepsilon^2\mathbf{I})^{-1}Y - \frac{n}{2}\log 2\pi. \tag{7.28}$$

Note that this approach is fairly general. It is even possible to introduce different kernel widths for each dimension, calculating the gradient with respect to all parameters, and performing the maximization by gradient ascent on parameter space in an efficient fashion. For our application, we will restrict ourselves to a single kernel width for all directions and perform an exhaustive search.

7.4.3 Computational Complexity Considerations

With regard to the computational complexity, both methods can be combined with kernel ridge regression such that the overall time complexity is $O(n^3)$. The computationally most demanding main step is to the compute eigendecomposition of $\mathbf{K}$. The estimate of the cut-off dimension can be achieved in $O(n^2)$ (mostly dominated by computing the spectrum of Y). The same observation holds true for GPML and LOOCV, where one needs an initial $O(n^3)$ to obtain the spectral decomposition of $\mathbf{K}$, and then $O(n^2)$ to compute the errors or likelihoods for a fixed given candidate for the regularization constant.

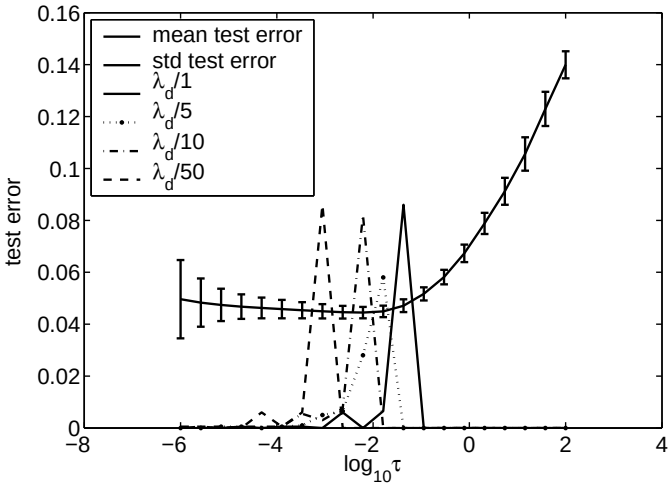


Figure 7.4: Different choices of ratio for selecting the regularization constant versus generalization error. The generalization error does not depend very strongly on the choice of the quotient, the values in the range from 5 to 10 lead to good results in this example.

7.5 Estimating the Regularization Parameter

Let us now turn to the question of estimating the regularization parameter for a fixed kernel based on the cut-off dimension estimators from Chapter 6. The idea is to adjust τ such that the resulting fit reconstructs the signal up to the cut-off dimension, discarding the noise.

In order to understand how this could be accomplished, recall the spectral decomposition of KRR as discussed in Section 7.3.2. The fit vector can be written as

$$\hat{f}(x) = \sum_{i=1}^{n} u_i^c(x) \frac{l_i}{l_i + \tau} u_i^\top Y, \tag{7.29}$$

where u_i^c is the Nyström extension of the eigenvector u_i (see (7.7)). As before, the scalar products $u_i^\top Y$ compute the coefficients of Y expressed in the basis $u_1, \ldots, u_n$. KRR then computes the fit by shrinking these coefficients by the shrinkage factors $w_i = l_i/(l_i + \tau)$, and reconstructing the resulting fit in the original basis.

As discussed in Section 7.3.3, as the eigenvalues tend to zero, the shrinkage weights vary from 1 to 0 in a non-increasing fashion. This change happens as the eigenvalues become smaller than τ and the weights decay as quickly as the eigenvalues, because $w_i \leq \min(l_i/\tau, 1)$.

We wish to set τ such that the shrinkage factor w_d is close to 1 at the cut-off point d and starts to decay for larger indices. The actual rate of decay will of course depend on the eigenvalues of the kernel matrix. We therefore propose to adjusting τ such that the weights will be close to 1 for $1 \leq i \leq d$. Let ϱ be a threshold, and we require that $w_d > \varrho$. This leads to the choice

$$\varrho = w_d = \frac{l_d}{l_d + \tau} \quad \Rightarrow \quad \tau = \frac{1 - \varrho}{\varrho} l_d. \tag{7.30}$$

The choice of ϱ is rather arbitrary, but the method itself is not very sensitive to this choice. In Figure 7.4, the distribution of the estimated regularization parameters τ for the noisy sinc example are plotted for the choices $\varrho \in \{1/2, 1/6, 1/10, 1/51\}$, leading to setting $\tau \in \{l_d, l_d/5, l_d/10, l_d/50\}$. We see that all choices are rather reasonable, but that $\tau = l_d$ leads to a slight underfit. Based on this example and further extensive experimental experience, we have found that $\varrho = 10/11$ works quite well in practice. We will call the method of first estimating the cut-off dimension and then setting the regularization parameter according to (7.30) the *spectrum method*.

It should be stressed that the heuristic nature of our approach is due to the lack of further *a priori* information about the data. More principled approaches necessarily rely on assumptions

The spectrum method for model selection.

Input:	kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$,
	labels $Y = (Y_1, \ldots, Y_n)^\top \in \mathbb{R}^n$.
Parameters:	lower threshold τ_0,
	default large regularization constant τ_1.
Output:	regularization constant τ

1 compute eigendecomposition $\mathbf{K} = \mathbf{ULU}^\top$ with
$\mathbf{L} = \operatorname{diag}(l_1, \ldots, l_n)$, $l_1 \geq \ldots \geq l_n$.

2 $s \leftarrow \mathbf{U}^T Y$.

3 for $j = 1, \ldots, \lceil n/2 \rceil$,

3a $$\sigma_1^2 \leftarrow \frac{1}{j} \sum_{i=1}^{j} s_i^2, \quad \sigma_2^2 \leftarrow \frac{1}{n-j} \sum_{i=j+1}^{n} s_i^2,$$

3b $$p_j \leftarrow \frac{j}{n} \log \sigma_1^2 + \frac{n-j}{n} \log \sigma_2^2.$$

4 set $d \leftarrow \operatorname*{argmin}_{j=1,\ldots,\lceil n/2 \rceil} p_j$
set $\tau \leftarrow l_d/10$.

5 *Guard against small regularization constants*
if $\tau < \tau_0$, set $\tau = \tau_1$.

6 return τ.

Figure 7.5: Estimating the cut-off dimension given a kernel matrix and a label vector.

which render the derivation of the procedure possible, but at the same time restrict the class of data sets for which the procedure is valid.

Finally, in high noise level situations, the two component model estimate of the cut-off dimension can become instable and estimate a very large cut-off dimension, such that the regularization constant becomes very small (smaller than 10^{-10}). Using such a regularization constant for the subsequent training step can result in numerical instabilities. Therefore, if the estimated τ is below a certain threshold, we assume that we are in a high-noise situation and set τ to a large value to essentially estimate the mean through the noise. In our experiments (see below), this case only occurred during the high noise settings in the noisy sinc function data set in Section 7.6. There, we used a threshold of $\tau_0 = 10^{-16}$ and set $\tau_1 = 10$ in that case.

Figure 7.5 summarizes the spectrum method used in conjunction with the two-component method for estimating the cut-off dimension, which can be found in Section 6.5.2. We prefer this method as it has proven to be both more efficient and robust than the alternative method based on a kernel density estimate and a resampling approach.

The computationally most expensive step is the computation of the spectral decomposition of the kernel matrix $\mathbf{K}$. Using one of the standard methods (see for example Golub and van Loan (1996)), this step has time complexity of $O(n^3)$. Naive computation of p_j would cost $O(n)$ for each j, but by not computing the sums from scratch for each j but rather using two accumulators, the computation time can be decreased to $O(1)$, leading to $O(n)$ for the remainder of the algorithm.

The spectrum method not only returns an estimate for τ but naturally also provides an estimate of the cut-off dimension of the data source. This estimate can provide useful additional information about the data set. The cut-off dimension can also be used to estimate the variance of the noise, because the spectrum without the leading coefficients up to the cut-off dimensions are assume to contain only noise. Therefore, the σ_2^2 computed in line 3a in Algorithm 7.5 is an estimate of the noise variance.

Both the cut-off dimension and the noise can give information concerning the hardness of the problem and the goodness of the fit apart from the mere training error. The training error alone provides no information on how close the fit is to the target function. On the other hand, if the cut-off dimension is small with respect to the number of samples, the noise will be removed almost

completely and the fit should be close to the target function. Since the minimal error is given by the noise variance, the noise variance provides a further indication of how good the fit is. In summary, if the cut-off dimension is small compared to the sample size and the error is close to the estimated noise variance, then the fit is presumably as good as possible. Note however, that different kernels realize different cut-off dimensions, such that it might be possible to achieve a superior fit with an alternative kernel on a given data set.

7.6 Regression Experiments

We compare the spectrum method to other state-of-the-art methods in a regression setting. We consider a toy data set given by the noisy sinc function, and several regression benchmark data sets.

7.6.1 Toy data set

We begin our regression experiments by the noisy sinc function data set. The following algorithms will be compared: leave-one-out cross-validation (LOOCV), maximum likelihood approach from the Gaussian process framework (GPML), and kernel ridge regression with the spectrum method (KRRSM). In these simulations, we will use the methods only to determine the regularization constant τ.

We tested the algorithms on the noisy sinc data set for varying choices of the parameters. We take the kernel width w and the noise standard deviations from the sets

$$w \in \{0.1, 0.3, 0.6, 1.0, 2.0, 5.0\},$$
$$\sigma_\varepsilon \in \{0.1, 0.3, 0.5, 1.0\}.$$

The noise levels of $\sigma_\varepsilon = 0.5$ and 1.0 are already rather severe. For $\sigma_\varepsilon = 1.0$, the sinc function is barely visible. For each combination of the parameters w and σ_ε, we generated a data set of size 1000. This data set is split 100 times into a training set of size 100 and a validation set of size 900. On each of these splits, for the given kernel widths, the algorithms estimate the regularization constant τ and train a kernel ridge regression solution with the final τ. Then, the generalization error is estimated on the validation set. We thus obtain for each algorithm and each parameter setting 100 generalization errors.

In Figure 7.6(a), the resulting errors are plotted together with their standard deviations. Note that the noise variance is equal to the optimal generalization error. Therefore, we know that these are given by $\sigma_\varepsilon^2 \in \{0.01, 0.09, 0.25, 1\}$. We see that the smallest errors for varying kernel width w are in fact close to the noise variance, such that the regression works well in the best case.

Now with respect to the different methods, first of all, we see that the methods perform comparably for most of the parameter settings. In fact, the only significant differences occur for kernel widths $w = 2.0$ and 5.0 and at noise level $\sigma_\varepsilon = 1.0$.

For kernel width $w = 2.0$, the GPML method consistently leads to higher test errors, while KRRSM is still close to the results of LOOCV. For $w = 5.0$, both GPML and KRRSM are worse than the results from LOOCV. We can get a hint for this effect from the estimated regularization constants which can be found in Figure 7.6(b). When comparing the estimated τs, we see that for $w \in \{0.1, 0.3, 0.6, 1.0\}$, GPML estimates the τ such they are appear to be estimates of the noise variance σ_ε^2. This observation is not surprising, because in the Gaussian process model, the regularization constant τ is equivalent to the noise variance (see (7.27)). Actually, setting τ to the (estimated) noise variance is the canonical choice of the regularization parameter τ in a Bayesian framework. We see from the experiments that depending on the kernel width w, this choice might not be justified (and in fact, it is also not justified from a theoretical point of view; if the kernel matrix does not reflect the true covariance structure of the Gaussian process, we cannot expect that adjusting τ alone leads to a good fit). For $w = 2.0$ and 5.0, we see from the results for LOOCV that much smaller regularization constants are needed to obtain a good fit. KRRSM achieves this

for $w = 2.0$, but proposes even smaller regularization constants for $w = 5.0$. In summary, for large kernel widths, GPML has the tendency to underfit, while KRRSM overfits slightly.

Finally for $\sigma_\varepsilon = 1.0$, KRRSM becomes a bit unstable which can be seen from the larger standard deviations of the test error over the 100 resamples. Looking at the proposed regularization constants, we see that every method seems to follow a different strategy: LOOCV estimates a rather large τ leading to a smooth fit of the data, while GPML estimates the noise variance. Finally, KRRSM proposes a similar τ as GPML but with much larger variance. To the defense of KRRSM, we should stress that the data sets at $\sigma_\varepsilon = 1.0$ contain almost only noise. Such data sets will be very rare in realistic settings. One would rather first work towards finding more meaningful features before applying a kernel method to such a data set and expect good results.

In summary, KRRSM works competitively with LOOCV at normal noise levels. It performed a bit worse for very high noise levels and at large kernel widths, but still better than GPML. Again, these results were obtained for fixed kernel widths. Finally, Figure 7.7 contains the estimated cut-off dimensions for KRRSM. We see that these are mostly constant for medium noise levels and become more instable for higher noise levels. We also see that the problem is not very hard. Having approximately 9 relevant dimension for 100 sample points should roughly allow to suppress the noise to a tenth of the original variance.

Next, we included the kernel widths into the parameters to be estimated by the methods. For all methods, the candidate kernel widths were 40 logarithmically spaces points between 10^{-2} and 10^4. For LOOCV, the leave-one-out cross-validation error was evaluated for all pairs of candidate kernel widths and regularization constants, and the parameter set leading to the smallest validation error was selected. For KRRSM, for each kernel width, τ was determined by the spectrum method. Then, the validation error was computed. The kernel width and regularization constant with the smallest validation error was selected. For GPML, the log-likelihoods were computed for each parameter set and the parameter maximizing the log-likelihood was selected. The algorithms were again evaluated on 100 resamples as in the last experiment.

Figure 7.8 plots the test errors. We see that KRRSM performs competitively with the other two methods. We also see that GPML consistently estimates τ to match the noise variance, which is in accordance with the interpretation of τ in the Gaussian process framework. Note that all algorithms have spots where they are instable resulting in large standard deviations in the estimated regularization constants or kernel widths.

We conclude that for the noisy sinc data set, KRRSM performs competitively with the state-of-the-art methods GPML and LOOCV. For fixed kernel widths, it is also more robust than GPML with respect to the choice of the wrong kernel width.

7.6.2 Benchmark Data Sets

The next question is how KRRSM performs on real world benchmark data sets. We therefore compare the algorithms on the `kin-8` and `bank-8` data sets from the DELVE repository[1]. The `bank` dataset is generated from a simple model of bank customers. Each customer has his own level of patience. The output variable to be predict is the percentage of rejected customers who were turn away before arriving at the head of the queues. The `kin` data set is generated from a simulator of an 8-joint robot arm. The task is to predict the position of the arm given the angles at the 8 joints. The noise is injected not only on the position measurement, but on the angle measurements, such that the noise is also transformed through the geometry of the robot arm leading to non-i.i.d. noise. These data sets come in four flavors, `nm`, `fm`, `nh`, and `fh`, where `f` stands for fairly linear, `n` for nonlinear concerning the dependency between the X and Y, and `m` for moderate noise and `h` for high-noise. We also use a further variant called `kin40k` prepared by Anton Schwaighofer[2] which has an even higher noise level.

[1] available from `http://www.cs.toronto.edu/~delve`

[2] available from `http://www.cis.tugraz.at/igi/aschwaig/data.html`.

	Method	$w = 0.1$	$w = 0.3$	$w = 0.6$	$w = 1.0$	$w = 2.0$	$w = 5.0$
$\sigma_\varepsilon = 0.1$	LOOCV	$1.17_{\pm 0.08} \times 10^{-2}$	$1.19_{\pm 0.62} \times 10^{-2}$	$1.12_{\pm 0.11} \times 10^{-2}$	$1.13_{\pm 0.09} \times 10^{-2}$	$1.13_{\pm 0.08} \times 10^{-2}$	$2.04_{\pm 0.13} \times 10^{-2}$
	GPML	$1.29_{\pm 0.13} \times 10^{-2}$	$1.15_{\pm 0.07} \times 10^{-2}$	$1.10_{\pm 0.05} \times 10^{-2}$	$1.14_{\pm 0.07} \times 10^{-2}$	$3.75_{\pm 0.22} \times 10^{-2}$	$7.75_{\pm 0.31} \times 10^{-2}$
	KRRSM	$1.16_{\pm 0.07} \times 10^{-2}$	$1.11_{\pm 0.05} \times 10^{-2}$	$1.10_{\pm 0.06} \times 10^{-2}$	$1.12_{\pm 0.06} \times 10^{-2}$	$1.24_{\pm 1.17} \times 10^{-2}$	$1.24_{\pm 0.05} \times 10^{-1}$
$\sigma_\varepsilon = 0.3$	LOOCV	$9.59_{\pm 0.55} \times 10^{-2}$	$9.37_{\pm 0.57} \times 10^{-2}$	$9.43_{\pm 0.69} \times 10^{-2}$	$9.53_{\pm 0.92} \times 10^{-2}$	$9.86_{\pm 1.81} \times 10^{-2}$	$1.05_{\pm 0.07} \times 10^{-1}$
	GPML	$1.03_{\pm 0.07} \times 10^{-1}$	$9.53_{\pm 0.52} \times 10^{-2}$	$9.25_{\pm 0.47} \times 10^{-2}$	$9.91_{\pm 0.47} \times 10^{-2}$	$1.32_{\pm 0.06} \times 10^{-1}$	$1.63_{\pm 0.07} \times 10^{-1}$
	KRRSM	$9.65_{\pm 0.50} \times 10^{-2}$	$9.42_{\pm 0.57} \times 10^{-2}$	$9.42_{\pm 0.72} \times 10^{-2}$	$9.54_{\pm 1.31} \times 10^{-2}$	$1.02_{\pm 0.24} \times 10^{-1}$	$1.99_{\pm 0.34} \times 10^{-1}$
$\sigma_\varepsilon = 0.5$	LOOCV	$2.91_{\pm 1.12} \times 10^{-1}$	$2.77_{\pm 0.15} \times 10^{-1}$	$2.82_{\pm 0.20} \times 10^{-1}$	$2.87_{\pm 0.22} \times 10^{-1}$	$2.88_{\pm 0.18} \times 10^{-1}$	$3.03_{\pm 0.21} \times 10^{-1}$
	GPML	$2.93_{\pm 0.18} \times 10^{-1}$	$2.78_{\pm 0.14} \times 10^{-1}$	$2.74_{\pm 0.13} \times 10^{-1}$	$2.85_{\pm 0.12} \times 10^{-1}$	$3.12_{\pm 0.11} \times 10^{-1}$	$3.29_{\pm 0.09} \times 10^{-1}$
	KRRSM	$2.91_{\pm 0.33} \times 10^{-1}$	$2.83_{\pm 0.22} \times 10^{-1}$	$2.87_{\pm 0.24} \times 10^{-1}$	$2.91_{\pm 0.32} \times 10^{-1}$	$3.00_{\pm 0.29} \times 10^{-1}$	$3.43_{\pm 0.15} \times 10^{-1}$
$\sigma_\varepsilon = 1.0$	LOOCV	$1.07_{\pm 0.06} \times 10^{0}$	$1.08_{\pm 0.09} \times 10^{0}$	$1.10_{\pm 0.08} \times 10^{0}$	$1.12_{\pm 0.12} \times 10^{0}$	$1.13_{\pm 0.10} \times 10^{0}$	$1.14_{\pm 0.15} \times 10^{0}$
	GPML	$1.11_{\pm 0.05} \times 10^{0}$	$1.07_{\pm 0.04} \times 10^{0}$	$1.06_{\pm 0.03} \times 10^{0}$	$1.08_{\pm 0.03} \times 10^{0}$	$1.10_{\pm 0.03} \times 10^{0}$	$1.12_{\pm 0.03} \times 10^{0}$
	KRRSM	$1.19_{\pm 0.21} \times 10^{0}$	$1.28_{\pm 1.46} \times 10^{0}$	$1.13_{\pm 0.13} \times 10^{0}$	$1.14_{\pm 0.18} \times 10^{0}$	$1.14_{\pm 0.11} \times 10^{0}$	$1.14_{\pm 0.23} \times 10^{0}$

(a) The test error.

	Method	$w = 0.1$	$w = 0.3$	$w = 0.6$	$w = 1.0$	$w = 2.0$	$w = 5.0$
$\sigma_\varepsilon = 0.1$	LOOCV	$3.17_{\pm 1.26} \times 10^{-1}$	$1.44_{\pm 0.76} \times 10^{-1}$	$2.40_{\pm 1.28} \times 10^{-2}$	$1.75_{\pm 1.43} \times 10^{-3}$	$6.86_{\pm 29.01} \times 10^{-6}$	$1.04_{\pm 0.14} \times 10^{-6}$
	GPML	$1.06_{\pm 0.24} \times 10^{-2}$	$9.71_{\pm 2.33} \times 10^{-3}$	$9.90_{\pm 2.36} \times 10^{-3}$	$1.08_{\pm 0.23} \times 10^{-2}$	$3.83_{\pm 0.90} \times 10^{-2}$	$7.62_{\pm 1.35} \times 10^{-2}$
	KRRSM	$4.37_{\pm 1.63} \times 10^{-1}$	$1.67_{\pm 0.52} \times 10^{-1}$	$3.41_{\pm 0.65} \times 10^{-2}$	$4.29_{\pm 2.03} \times 10^{-3}$	$5.19_{\pm 19.60} \times 10^{-6}$	$6.54_{\pm 5.13} \times 10^{-10}$
$\sigma_\varepsilon = 0.3$	LOOCV	$1.22_{\pm 0.45} \times 10^{0}$	$6.08_{\pm 3.34} \times 10^{-1}$	$1.03_{\pm 0.85} \times 10^{-1}$	$1.00_{\pm 0.93} \times 10^{-2}$	$1.24_{\pm 1.42} \times 10^{-4}$	$1.09_{\pm 0.27} \times 10^{-6}$
	GPML	$9.51_{\pm 2.18} \times 10^{-2}$	$9.66_{\pm 2.27} \times 10^{-2}$	$9.69_{\pm 2.23} \times 10^{-2}$	$1.05_{\pm 0.25} \times 10^{-1}$	$1.39_{\pm 0.23} \times 10^{-1}$	$1.77_{\pm 0.41} \times 10^{-1}$
	KRRSM	$5.37_{\pm 1.62} \times 10^{-1}$	$2.54_{\pm 1.39} \times 10^{-1}$	$4.35_{\pm 4.15} \times 10^{-2}$	$5.46_{\pm 5.20} \times 10^{-3}$	$1.02_{\pm 0.79} \times 10^{-4}$	$9.14_{\pm 17.32} \times 10^{-8}$
$\sigma_\varepsilon = 0.5$	LOOCV	$2.51_{\pm 1.04} \times 10^{0}$	$1.57_{\pm 0.84} \times 10^{0}$	$4.15_{\pm 3.64} \times 10^{-1}$	$7.10_{\pm 11.37} \times 10^{-2}$	$7.16_{\pm 55.41} \times 10^{-3}$	$3.37_{\pm 32.56} \times 10^{-4}$
	GPML	$2.69_{\pm 0.64} \times 10^{-1}$	$2.75_{\pm 0.65} \times 10^{-1}$	$2.78_{\pm 0.65} \times 10^{-1}$	$3.00_{\pm 0.62} \times 10^{-1}$	$3.41_{\pm 0.36} \times 10^{-1}$	$3.73_{\pm 0.74} \times 10^{-1}$
	KRRSM	$7.58_{\pm 3.12} \times 10^{-1}$	$4.01_{\pm 2.64} \times 10^{-1}$	$1.19_{\pm 0.80} \times 10^{-1}$	$2.21_{\pm 2.68} \times 10^{-2}$	$8.15_{\pm 19.28} \times 10^{-4}$	$7.01_{\pm 48.84} \times 10^{-7}$
$\sigma_\varepsilon = 1.0$	LOOCV	$1.37_{\pm 1.95} \times 10^{1}$	$1.41_{\pm 2.35} \times 10^{1}$	$1.56_{\pm 2.87} \times 10^{1}$	$1.46_{\pm 2.88} \times 10^{1}$	$1.66_{\pm 3.25} \times 10^{1}$	$1.69_{\pm 3.37} \times 10^{1}$
	GPML	$8.91_{\pm 1.15} \times 10^{-1}$	$8.89_{\pm 0.96} \times 10^{-1}$	$8.95_{\pm 1.10} \times 10^{-1}$	$9.14_{\pm 1.38} \times 10^{-1}$	$9.55_{\pm 1.86} \times 10^{-1}$	$9.71_{\pm 2.02} \times 10^{-1}$
	KRRSM	$7.79_{\pm 5.55} \times 10^{-1}$	$6.49_{\pm 7.82} \times 10^{-1}$	$6.60_{\pm 10.05} \times 10^{-1}$	$6.05_{\pm 11.72} \times 10^{-1}$	$6.82_{\pm 14.94} \times 10^{-1}$	$5.56_{\pm 17.14} \times 10^{-1}$

(b) The estimated τ.

Figure 7.6: Results for the noisy sinc function data set. Kernel widths are kept fixed, regularization constants are estimated.

	$w = 0.1$	$w = 0.3$	$w = 0.6$	$w = 1.0$	$w = 2.0$	$w = 5.0$
$\sigma_\varepsilon = 0.1$	$9_{\pm 2}$	$9_{\pm 1}$	$9_{\pm 0}$	$9_{\pm 1}$	$11_{\pm 0}$	$11_{\pm 1}$
$\sigma_\varepsilon = 0.3$	$8_{\pm 2}$	$9_{\pm 1}$	$9_{\pm 1}$	$9_{\pm 1}$	$9_{\pm 2}$	$11_{\pm 1}$
$\sigma_\varepsilon = 0.5$	$7_{\pm 4}$	$7_{\pm 4}$	$7_{\pm 4}$	$9_{\pm 4}$	$9_{\pm 3}$	$9_{\pm 4}$
$\sigma_\varepsilon = 1.0$	$6_{\pm 13}$	$8_{\pm 13}$	$8_{\pm 10}$	$9_{\pm 12}$	$9_{\pm 12}$	$10_{\pm 12}$

Figure 7.7: The cut-off dimension estimated by KRRSM.

		LOOCV	GPML	KRRSM
$\sigma_\varepsilon = 0.1$	error =	$1.05_{\pm 0.09} \times 10^{-2}$	$1.01_{\pm 0.05} \times 10^{-2}$	$1.04_{\pm 0.08} \times 10^{-2}$
	w =	$9.28_{\pm 5.68} \times 10^{-1}$	$6.72_{\pm 0.73} \times 10^{-1}$	$1.02_{\pm 0.75} \times 10^{0}$
	τ =	$2.54_{\pm 5.11} \times 10^{-2}$	$8.82_{\pm 1.95} \times 10^{-3}$	$4.62_{\pm 8.87} \times 10^{-2}$
$\sigma_\varepsilon = 0.3$	error =	$1.10_{\pm 0.13} \times 10^{-1}$	$1.07_{\pm 0.05} \times 10^{-1}$	$1.11_{\pm 0.15} \times 10^{-1}$
	w =	$6.83_{\pm 6.55} \times 10^{-1}$	$5.90_{\pm 1.10} \times 10^{-1}$	$9.48_{\pm 10.51} \times 10^{-1}$
	τ =	$4.34_{\pm 4.25} \times 10^{-1}$	$9.24_{\pm 1.91} \times 10^{-2}$	$2.20_{\pm 2.62} \times 10^{-1}$
$\sigma_\varepsilon = 0.5$	error =	$3.20_{\pm 2.04} \times 10^{-1}$	$2.92_{\pm 0.21} \times 10^{-1}$	$3.04_{\pm 0.40} \times 10^{-1}$
	w =	$7.20_{\pm 9.95} \times 10^{-1}$	$1.12_{\pm 10.06} \times 10^{2}$	$1.16_{\pm 1.34} \times 10^{0}$
	τ =	$1.16_{\pm 1.02} \times 10^{0}$	$2.55_{\pm 0.62} \times 10^{-1}$	$3.20_{\pm 4.81} \times 10^{-1}$
$\sigma_\varepsilon = 1.0$	error =	$1.07_{\pm 0.16} \times 10^{0}$	$1.08_{\pm 0.04} \times 10^{0}$	$1.12_{\pm 0.19} \times 10^{0}$
	w =	$8.01_{\pm 27.26} \times 10^{2}$	$4.12_{\pm 4.77} \times 10^{3}$	$1.62_{\pm 3.58} \times 10^{3}$
	τ =	$1.56_{\pm 2.44} \times 10^{1}$	$9.66_{\pm 1.97} \times 10^{-1}$	$3.13_{\pm 4.53} \times 10^{1}$

Figure 7.8: Test errors for the noisy sinc dataset. Kernel widths and regularization constants are selected are estimated.

Each data set is split into 100 realizations of training and test set, where the training set has size 100 and the test set contains the remaining data points. For the `kin` and `bank` data sets, the test set has size 8092, and it has size 39000 for `kin40k` .

The `kin` and `bank` data sets are both 8 dimensional. For preprocessing, the input dimensions have been scaled such that the input vectors are contained in $[-1, 1]^8$. The output variable has not been altered. The `kin40k` data set is used in its original form.

As candidate values for the kernel width w we used 40 logarithmically spaces values from 10^{-2} to 10^4. For τ, we used 40 logarithmically spaces values from 10^{-6} to 10^2.

Figure 7.9 shows the test errors for the data sets, while Figure 7.11 shows the estimated kernel widths and Figure 7.12 contains the estimated regularization constants. The cut-off dimension estimated by KRRSM are summarized in Figure 7.14.

The test errors are very similar for all methods, also with respect to the standard deviation over the resamples, with the following exceptions: GPML performs slightly worse on the `bank-8fh` data set, while KRRSM performs not on par for `kin-8fm`. Looking at the estimated parameters we see that on `kin-8fm`, KRRSM suggest roughly the same kernel width as LOOCV, but a regularization constant which is 100 times as large as LOOCV. Therefore, KRRSM underfits `kin-8fm`. This effect is due to the fact that the estimated dimension of 9 is too small. The means and standard deviations of the spectrum over all 100 realizations of the data are plotted in Figure 7.10. We see that there is strong evidence for the signal to be contained in the first 9 dimensions. On the other hand, up to $i = 40$, the spectrum decays on a slow ramp, which is not recognized by KRRSM as such. This either indicates very non-i.i.d. noise or a non-smooth target function. Given how the `kin` data sets are generated, it seems that KRRSM performs suboptimally because the noise is non-i.i.d and does not lead to a flat spectrum. On the other hand, it seems that estimating the cut-off dimension at 40 leads to a smaller τ and thus to a better fit.

Apart from that, the parameters estimated by KRRSM are similar to those of LOOCV. GPML again estimates the τ according to the noise levels. Estimates for the noise levels are obtained from KRRSM by measuring the variance of the spectrum beyond the cut-off point. These are plotted in Figure 7.13, although these numbers have to be considered with a certain precaution given the

Dataset	LOOCV	GPML	KRRSM
bank-8fh-boxed	$6.02_{\pm 0.48}\times 10^{-3}$	$6.51_{\pm 0.52}\times 10^{-3}$	$6.04_{\pm 0.50}\times 10^{-3}$
bank-8fm-boxed	$1.88_{\pm 0.39}\times 10^{-3}$	$1.90_{\pm 0.18}\times 10^{-3}$	$1.85_{\pm 0.23}\times 10^{-3}$
bank-8nh-boxed	$3.92_{\pm 0.44}\times 10^{-3}$	$3.88_{\pm 0.36}\times 10^{-3}$	$3.90_{\pm 0.40}\times 10^{-3}$
bank-8nm-boxed	$1.05_{\pm 0.21}\times 10^{-3}$	$9.99_{\pm 1.00}\times 10^{-4}$	$1.06_{\pm 0.26}\times 10^{-3}$
kin-8fh-boxed	$2.21_{\pm 0.14}\times 10^{-3}$	$2.23_{\pm 0.11}\times 10^{-3}$	$2.23_{\pm 0.11}\times 10^{-3}$
kin-8fm-boxed	$2.46_{\pm 0.25}\times 10^{-4}$	$2.46_{\pm 0.32}\times 10^{-4}$	$3.90_{\pm 1.09}\times 10^{-4}$
kin-8nh-boxed	$5.00_{\pm 0.44}\times 10^{-2}$	$4.96_{\pm 0.32}\times 10^{-2}$	$4.99_{\pm 0.51}\times 10^{-2}$
kin-8nm-boxed	$3.94_{\pm 0.44}\times 10^{-2}$	$4.03_{\pm 0.32}\times 10^{-2}$	$4.00_{\pm 0.46}\times 10^{-2}$
kin40k	$3.72_{\pm 0.34}\times 10^{-2}$	$3.93_{\pm 0.30}\times 10^{-2}$	$3.92_{\pm 0.46}\times 10^{-2}$

Figure 7.9: Mean square test errors for the regression benchmark data sets.

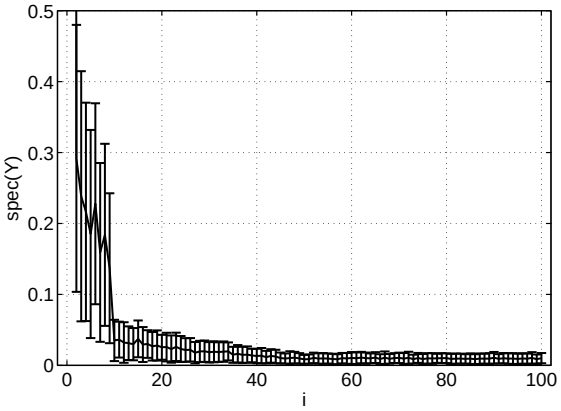


Figure 7.10: Spectrum for the `kin8fm` data set.

nature of the noise (see above discussion for the `kin` data sets). We see that these are very similar to the τ estimates of GPML. Consequently, the kernel widths estimated by GPML are different from those of both LOOCV and KRRSM. It is interesting to see that the resulting training errors are very similar. Moreover, since the optimal generalization error is given by noise variance, the estimates for the noise variance give some indication of how good the fit is compared to the optimum. With the exception of `kin-8fm`, the estimated variance are in fact all slightly smaller than the achieved fit, even for all methods. This can be attributed to the effect of generalization to new points and the small sample size.

Finally, from the cut-off dimensions in Figure 7.14, we see that the data sets are moderately complex and the 100 sample points used should already lead to fairly good results.

Dataset	LOOCV	GPML	KRRSM
bank-8fh-boxed	$5.86_{\pm 4.45}\times 10^{3}$	$1.72_{\pm 0.28}\times 10^{1}$	$5.84_{\pm 4.78}\times 10^{3}$
bank-8fm-boxed	$4.09_{\pm 4.36}\times 10^{3}$	$1.54_{\pm 0.24}\times 10^{1}$	$3.35_{\pm 4.62}\times 10^{3}$
bank-8nh-boxed	$4.02_{\pm 4.30}\times 10^{3}$	$1.84_{\pm 10.20}\times 10^{2}$	$4.10_{\pm 4.74}\times 10^{3}$
bank-8nm-boxed	$1.33_{\pm 2.34}\times 10^{3}$	$4.29_{\pm 1.82}\times 10^{1}$	$2.87_{\pm 4.38}\times 10^{3}$
kin-8fh-boxed	$1.22_{\pm 2.58}\times 10^{3}$	$4.61_{\pm 1.46}\times 10^{2}$	$8.33_{\pm 25.44}\times 10^{2}$
kin-8fm-boxed	$1.21_{\pm 1.49}\times 10^{2}$	$1.22_{\pm 0.29}\times 10^{2}$	$8.37_{\pm 9.30}\times 10^{1}$
kin-8nh-boxed	$8.19_{\pm 21.07}\times 10^{2}$	$8.96_{\pm 6.69}\times 10^{1}$	$7.24_{\pm 25.59}\times 10^{2}$
kin-8nm-boxed	$2.68_{\pm 10.68}\times 10^{2}$	$3.23_{\pm 3.35}\times 10^{1}$	$1.12_{\pm 9.99}\times 10^{2}$
kin40k	$2.12_{\pm 12.06}\times 10^{2}$	$6.20_{\pm 6.15}\times 10^{1}$	$3.17_{\pm 17.12}\times 10^{2}$

Figure 7.11: Estimated kernel widths for the regression benchmark data.

Dataset	LOOCV	GPML	KRRSM
bank-8fh-boxed	$4.29_{\pm 19.89}\times 10^{-4}$	$5.09_{\pm 1.24}\times 10^{-3}$	$3.90_{\pm 9.79}\times 10^{-4}$
bank-8fm-boxed	$5.76_{\pm 18.70}\times 10^{-4}$	$1.08_{\pm 0.27}\times 10^{-3}$	$8.79_{\pm 12.56}\times 10^{-4}$
bank-8nh-boxed	$6.66_{\pm 25.04}\times 10^{-4}$	$3.20_{\pm 1.09}\times 10^{-3}$	$9.31_{\pm 91.99}\times 10^{-2}$
bank-8nm-boxed	$1.71_{\pm 6.70}\times 10^{-4}$	$6.09_{\pm 1.94}\times 10^{-4}$	$8.45_{\pm 13.43}\times 10^{-4}$
kin-8fh-boxed	$1.02_{\pm 1.70}\times 10^{-2}$	$1.95_{\pm 0.42}\times 10^{-3}$	$3.81_{\pm 2.86}\times 10^{-2}$
kin-8fm-boxed	$4.21_{\pm 3.58}\times 10^{-4}$	$1.56_{\pm 0.36}\times 10^{-4}$	$2.52_{\pm 2.00}\times 10^{-2}$
kin-8nh-boxed	$1.47_{\pm 1.45}\times 10^{-1}$	$4.29_{\pm 1.01}\times 10^{-2}$	$4.89_{\pm 12.03}\times 10^{-1}$
kin-8nm-boxed	$7.64_{\pm 9.18}\times 10^{-2}$	$2.80_{\pm 0.93}\times 10^{-2}$	$3.37_{\pm 6.85}\times 10^{-1}$
kin40k	$5.18_{\pm 6.85}\times 10^{-2}$	$2.57_{\pm 1.04}\times 10^{-2}$	$3.85_{\pm 8.15}\times 10^{-1}$

Figure 7.12: Estimated τ for the regression benchmark data.

Dataset	estimated noise variance
bank-8fh-boxed	$5.28_{\pm 1.10}\times 10^{-3}$
bank-8fm-boxed	$1.42_{\pm 0.32}\times 10^{-3}$
bank-8nh-boxed	$3.27_{\pm 0.87}\times 10^{-3}$
bank-8nm-boxed	$7.35_{\pm 2.56}\times 10^{-4}$
kin-8fh-boxed	$2.10_{\pm 0.46}\times 10^{-3}$
kin-8fm-boxed	$3.76_{\pm 0.78}\times 10^{-4}$
kin-8nh-boxed	$4.85_{\pm 0.98}\times 10^{-2}$
kin-8nm-boxed	$4.23_{\pm 0.88}\times 10^{-2}$
kin40k	$4.24_{\pm 0.89}\times 10^{-2}$

Figure 7.13: Estimated noise variance for the regression benchmark data.

Dataset	KRRSM
bank-8fh-boxed	$9_{\pm 2}$
bank-8fm-boxed	$10_{\pm 4}$
bank-8nh-boxed	$9_{\pm 3}$
bank-8nm-boxed	$15_{\pm 8}$
kin-8fh-boxed	$8_{\pm 2}$
kin-8fm-boxed	$9_{\pm 2}$
kin-8nh-boxed	$6_{\pm 3}$
kin-8nm-boxed	$8_{\pm 3}$
kin40k	$8_{\pm 4}$

Figure 7.14: Estimated cut-off dimension for the regression benchmark data.

In summary, KRRSM performs competitively to LOOCV and GPML. This result shows that the structural insights from Chapter 6 can actually be used to perform effective model selection for kernel ridge regression. Beyond the main task of estimating good parameters, KRRSM also computes an estimated dimensionality of the problem which provides further insights into the nature of the learning problem. Using the estimated cut-off dimension one can estimate the noise variance of the regression problem from the spectrum. This estimate also gives an indication for the optimal test error.

7.7 Classification Experiments

The spectrum method has been motivated and derived in a regression setting. In this section, we want to explore how good the methods work for classification tasks.

As usual, to apply a regression method to two-class classification problems, one class is labeled with 1 and the other with -1. With these conventions, the target function f is given as (see Section 2.3)

$$f(x) = \mathbf{P}\{Y = 1 \mid X = x\} - \mathbf{P}\{Y = -1 \mid X = X\} = \mathbf{E}(Y|X = x). \tag{7.31}$$

The noise is thus given as $Y - \mathbf{E}(Y \mid X = x)$, which has mean zero, but a discrete distribution.

7.7.1 Benchmark Data Sets

We use the benchmark data set from Rätsch et al. (2001), which consists of thirteen artificial and real world data sets. Each data set has already been split into 100 realizations of training and test data.

We compare the spectrum method to a tentative gold-standard achieved by a support vector machine (SVM) whose hyperparameters have been fine-tuned by exhaustive search and k-fold cross validation. The results for the SVM in Rätsch et al. (2001) were obtained as follows. The hyperparameters (regularization constant C and kernel width w) were first determined on the first five data sets. For each of these data sets, the parameters were estimated using 5-fold cross validation. Then, the median of these five estimates were taken for the evaluation on all 100 realizations of the data set.

For LOOCV and GPML, the following set of candidate kernel widths and regularization constants were used: the kernel widths were 20 logarithmically spaced values from 10^{-3} to 10^{3}, and the regularization constant τ was tested on 20 logarithmically spaces values from 10^{-6} to 10^{2}.

For KRRSM, the same set of kernel widths was used but for each width, τ was selected by the spectrum method. We also included a variant of the spectrum method based on the resampling based modality estimate besides the standard method based on the two component model. This will be called KRRRB.

Figure 7.15 plots the 0-1-loss test errors. We see that with the exception of KRRRB which consistently performs worse, GPML, KRRSM, and LOOCV compare similarly as the SVM. For the `banana` data set, GPML, KRRSM, and LOOCV even perform slightly better. On the `flare-solar` and `ringnorm` data set, the three methods perform worse than SVM, although for the `ringnorm` data set, KRRSM is still as good as LOOCV whereas GPML performs as bad as KRRRB. Finally, for the `twonorm` data set, KRRSM even achieves the best result.

The dimension estimated by KRRRB and KRRSM are summarized in Figure 7.18. These results support the observations from Chapter 6 that KRRRB is much more instable than KRRSM. The only exception is the `image` data set where KRRSM is much less stable than KRRRB, presumably because this is a low noise data set, which is indicated by the small achievable test error. Some of the data sets seem to have fairly high cut-off dimensions. On the other hand, the seemingly hardest data sets `breast-cancer`, `flare-solar`, and `titanic` have a small cut-off dimension. This suggests that the noise is rather large and that it is unlikely that better solutions can be found, at least with rbf-kernels. Another interesting value is the cut-off dimension of KRRSM for

Dataset	SVM	GPML	KRRRB	KRRSM	LOOCV
banana	$11.5_{\pm 0.7}$	$10.4_{\pm 0.4}$	$12.2_{\pm 3.4}$	$10.6_{\pm 0.5}$	$10.6_{\pm 0.6}$
breast-cancer	$26.0_{\pm 4.7}$	$27.2_{\pm 5.1}$	$33.2_{\pm 8.5}$	$26.4_{\pm 4.7}$	$26.6_{\pm 4.7}$
diabetis	$23.5_{\pm 1.7}$	$23.0_{\pm 1.7}$	$27.5_{\pm 2.8}$	$23.2_{\pm 1.7}$	$23.3_{\pm 1.7}$
flare-solar	$32.4_{\pm 1.8}$	$34.1_{\pm 1.7}$	$40.6_{\pm 9.1}$	$34.2_{\pm 1.8}$	$34.1_{\pm 1.8}$
german	$23.6_{\pm 2.1}$	$24.0_{\pm 2.2}$	$24.7_{\pm 2.3}$	$23.4_{\pm 2.3}$	$23.5_{\pm 2.2}$
heart	$16.0_{\pm 3.3}$	$18.4_{\pm 3.4}$	$17.9_{\pm 4.0}$	$16.4_{\pm 3.3}$	$16.6_{\pm 3.6}$
image	$3.0_{\pm 0.6}$	$2.8_{\pm 0.5}$	$2.7_{\pm 0.5}$	$2.7_{\pm 0.5}$	$2.8_{\pm 0.5}$
ringnorm	$1.7_{\pm 0.1}$	$6.0_{\pm 0.9}$	$6.1_{\pm 1.1}$	$4.9_{\pm 0.7}$	$4.9_{\pm 0.6}$
splice	$10.9_{\pm 0.6}$	$11.5_{\pm 0.6}$	$11.1_{\pm 0.7}$	$11.1_{\pm 0.6}$	$11.2_{\pm 0.7}$
thyroid	$4.8_{\pm 2.2}$	$4.3_{\pm 2.7}$	$4.6_{\pm 2.3}$	$4.4_{\pm 2.2}$	$4.4_{\pm 2.2}$
titanic	$22.4_{\pm 1.0}$	$22.7_{\pm 1.3}$	$25.3_{\pm 11.3}$	$22.4_{\pm 0.9}$	$22.4_{\pm 0.9}$
twonorm	$3.0_{\pm 0.2}$	$3.1_{\pm 0.2}$	$3.7_{\pm 0.4}$	$2.5_{\pm 0.1}$	$2.8_{\pm 0.2}$
waveform	$9.9_{\pm 0.4}$	$10.0_{\pm 0.5}$	$10.9_{\pm 0.7}$	$10.0_{\pm 0.4}$	$9.7_{\pm 0.4}$

Figure 7.15: Test error rates for the classification benchmark data sets. The data sets are from Rätsch et al. (2001) and are available online from `http://www.first.fraunhofer.de/~raetsch`.

Dataset	LOOCV	GPML	KRRRB	KRRSM
banana	$2.65_{\pm 1.08}\times 10^{-1}$	$3.34_{\pm 0.17}\times 10^{-1}$	$1.06_{\pm 4.65}\times 10^{0}$	$2.59_{\pm 1.29}\times 10^{-1}$
breast-cancer	$4.66_{\pm 16.93}\times 10^{1}$	$5.81_{\pm 3.04}\times 10^{1}$	$2.14_{\pm 3.76}\times 10^{2}$	$3.01_{\pm 9.92}\times 10^{1}$
diabetis	$1.52_{\pm 1.16}\times 10^{1}$	$2.61_{\pm 0.19}\times 10^{1}$	$4.16_{\pm 15.74}\times 10^{1}$	$4.80_{\pm 14.46}\times 10^{1}$
flare-solar	$1.79_{\pm 3.67}\times 10^{2}$	$5.78_{\pm 1.04}\times 10^{0}$	$8.99_{\pm 19.15}\times 10^{1}$	$2.58_{\pm 4.14}\times 10^{2}$
german	$2.83_{\pm 3.19}\times 10^{1}$	$1.01_{\pm 0.23}\times 10^{2}$	$5.29_{\pm 18.78}\times 10^{1}$	$2.72_{\pm 0.84}\times 10^{1}$
heart	$3.21_{\pm 3.86}\times 10^{2}$	$1.99_{\pm 1.35}\times 10^{1}$	$1.98_{\pm 3.14}\times 10^{2}$	$4.88_{\pm 4.65}\times 10^{2}$
image	$2.21_{\pm 0.79}\times 10^{0}$	$8.68_{\pm 5.40}\times 10^{-1}$	$2.44_{\pm 1.16}\times 10^{0}$	$2.29_{\pm 1.48}\times 10^{0}$
ringnorm	$2.54_{\pm 0.35}\times 10^{1}$	$1.27_{\pm 0.00}\times 10^{1}$	$1.59_{\pm 0.76}\times 10^{1}$	$2.34_{\pm 0.57}\times 10^{1}$
splice	$5.17_{\pm 0.87}\times 10^{1}$	$5.31_{\pm 0.63}\times 10^{1}$	$4.89_{\pm 1.16}\times 10^{1}$	$5.46_{\pm 0.00}\times 10^{1}$
thyroid	$2.92_{\pm 0.50}\times 10^{0}$	$4.44_{\pm 1.65}\times 10^{-1}$	$2.77_{\pm 1.09}\times 10^{0}$	$2.78_{\pm 0.52}\times 10^{0}$
titanic	$2.09_{\pm 3.94}\times 10^{2}$	$6.82_{\pm 6.59}\times 10^{0}$	$1.63_{\pm 2.98}\times 10^{2}$	$1.66_{\pm 3.67}\times 10^{2}$
twonorm	$1.33_{\pm 0.27}\times 10^{1}$	$2.92_{\pm 0.85}\times 10^{1}$	$1.34_{\pm 0.31}\times 10^{1}$	$1.34_{\pm 0.30}\times 10^{1}$
waveform	$1.58_{\pm 0.58}\times 10^{1}$	$2.48_{\pm 0.66}\times 10^{1}$	$1.47_{\pm 3.64}\times 10^{1}$	$1.55_{\pm 1.10}\times 10^{1}$

Figure 7.16: Estimated kernel widths for the classification benchmark data set.

the `twonorm` data set, which is $d = 2$. This means that the resulting best solution for the `twonorm` data set is computed by a very low complexity fit.

In summary, we have observed two behaviors: First of all, regression methods can be used very well for classification although they have been derived in a completely different setting. It is safe to assume that for GPML, none of the original assumptions are justified. This insight is not new but has already been stated in (Rifkin, 2002). Second of all, we have seen that KRRSM shows the same performance as LOOCV, in one case even being significantly better.

7.8 Conclusion

In this chapter, we have explored applications of the theoretical results achieved so far to kernel ridge regression. We have shown that kernel ridge regression practically works by first transforming the data into a representation where the noise can be removed by simply shrinking a number of coefficients to zero. Then, we have proposed a method for adjusting the regularization parameter based on the cut-off dimension estimators. Experimentally, we observed that this method performs very competitively to state-of-the-art methods.

Dataset	LOOCV	GPML	KRRRB	KRRSM
banana	$3.35_{\pm 2.21}\times 10^{-1}$	$2.98_{\pm 0.00}\times 10^{-1}$	$1.55_{\pm 1.68}\times 10^{-1}$	$2.90_{\pm 1.40}\times 10^{-1}$
breast-cancer	$1.96_{\pm 1.15}\times 10^{0}$	$7.77_{\pm 0.78}\times 10^{-1}$	$6.93_{\pm 4.17}\times 10^{-2}$	$1.14_{\pm 0.31}\times 10^{0}$
diabetis	$1.66_{\pm 0.66}\times 10^{0}$	$7.85_{\pm 0.00}\times 10^{-1}$	$1.15_{\pm 1.18}\times 10^{-1}$	$6.45_{\pm 2.77}\times 10^{-1}$
flare-solar	$5.14_{\pm 4.79}\times 10^{-1}$	$7.85_{\pm 0.00}\times 10^{-1}$	$4.49_{\pm 10.45}\times 10^{-2}$	$3.16_{\pm 4.65}\times 10^{-1}$
german	$1.73_{\pm 0.61}\times 10^{0}$	$7.85_{\pm 0.00}\times 10^{-1}$	$2.07_{\pm 3.63}\times 10^{-1}$	$1.05_{\pm 0.14}\times 10^{0}$
heart	$4.72_{\pm 4.34}\times 10^{-1}$	$3.16_{\pm 1.56}\times 10^{-1}$	$3.00_{\pm 4.07}\times 10^{-1}$	$4.19_{\pm 5.36}\times 10^{-1}$
image	$1.61_{\pm 0.73}\times 10^{-2}$	$2.09_{\pm 2.09}\times 10^{-2}$	$6.27_{\pm 2.93}\times 10^{-2}$	$4.22_{\pm 2.98}\times 10^{-2}$
ringnorm	$1.11_{\pm 0.10}\times 10^{-1}$	$4.07_{\pm 0.72}\times 10^{-2}$	$3.82_{\pm 1.66}\times 10^{-2}$	$7.71_{\pm 2.65}\times 10^{-2}$
splice	$1.00_{\pm 0.61}\times 10^{-1}$	$1.69_{\pm 4.14}\times 10^{-2}$	$6.75_{\pm 1.57}\times 10^{-2}$	$8.79_{\pm 0.76}\times 10^{-2}$
thyroid	$1.15_{\pm 0.72}\times 10^{-1}$	$2.03_{\pm 4.71}\times 10^{-3}$	$1.49_{\pm 0.83}\times 10^{-1}$	$9.83_{\pm 4.20}\times 10^{-2}$
titanic	$1.39_{\pm 2.20}\times 10^{0}$	$7.70_{\pm 0.84}\times 10^{-1}$	$5.53_{\pm 10.54}\times 10^{-2}$	$6.92_{\pm 6.80}\times 10^{-1}$
twonorm	$4.61_{\pm 2.42}\times 10^{-1}$	$1.11_{\pm 0.12}\times 10^{-1}$	$1.32_{\pm 4.67}\times 10^{-1}$	$2.61_{\pm 0.38}\times 10^{0}$
waveform	$7.60_{\pm 2.84}\times 10^{-1}$	$2.63_{\pm 0.73}\times 10^{-1}$	$9.42_{\pm 8.70}\times 10^{-2}$	$3.90_{\pm 1.34}\times 10^{-1}$

Figure 7.17: Estimated regularization constants for the classification benchmark data set.

Dataset	KRRRB	KRRSM	n
banana	$61_{\pm 58}$	$27_{\pm 5}$	400
breast-cancer	$100_{\pm 36}$	$3_{\pm 1}$	200
diabetis	$197_{\pm 76}$	$8_{\pm 1}$	468
flare-solar	$72_{\pm 50}$	$9_{\pm 2}$	666
german	$106_{\pm 65}$	$12_{\pm 1}$	700
heart	$7_{\pm 19}$	$4_{\pm 2}$	170
image	$200_{\pm 0}$	$266_{\pm 82}$	1300
ringnorm	$179_{\pm 42}$	$44_{\pm 14}$	400
splice	$200_{\pm 0}$	$86_{\pm 12}$	1000
thyroid	$12_{\pm 7}$	$15_{\pm 5}$	140
titanic	$8_{\pm 4}$	$6_{\pm 2}$	150
twonorm	$171_{\pm 56}$	$2_{\pm 0}$	400
waveform	$140_{\pm 56}$	$15_{\pm 6}$	400

Figure 7.18: Estimated cut-off dimension and training sample sizes for the classification benchmark data set.

In summary, the theoretical results from Chapters 3, 4 and 6 are useful to analyze kernel methods. The proposed methods also provided useful additional information about the data sets like the dimensionality of the problem and the amount of noise present in the data.

Chapter 8

Conclusion

This thesis presents a detailed analysis of the spectral structure of the kernel matrix, and applications of these results to machine learning algorithms. The kernel matrix defines a central component in virtually all kernel methods, such that detailed knowledge of the structure of the kernel matrix is of great use for both, theory and practice.

The theoretical analysis of the spectral properties of the kernel matrix were guided by the central concern that the resulting bounds actually match the behavior of the approximation errors as can be observed in numerical simulations. In such simulations, one can observe that small eigenvalues fluctuate much less than larger eigenvalues. Existing results seriously failed in reflecting this behavior, since the existing bounds did not depend on the eigenvalues in the right manner and did not scale appropriately.

The convergence results for the eigenvalues combined classical results from the perturbation theory of Hermitian matrices with probabilistic finite sample size bounds on the norm of certain error matrices to obtain a relative-absolute bound which is considerable tighter than bounds which existed so far. Compared to the approaches which were based on the Courant-Fisher variational characterization of the eigenvalues, the size of the true eigenvalue enters the bound quite naturally, leading to bounds which reflect the behavior of observed approximation errors. Being able to support this observation with a theoretical result has proved to be very valuable.

The basic relative-absolute bound is stated very generally with respect to several error matrices, which can be easily proven to imply convergence of the eigenvalues. It is slightly more intricate to obtain actual finite-sample size bounds on theses errors. We have undertaken this analysis for two cases: that of a Mercer kernel with uniformly bounded eigenfunctions, and for the case of kernel functions which are uniformly bounded, covering a range of relevant kernel functions, including, for example, the ubiquitous radial basis function kernels. These estimates showed that if the eigenvalues decay quickly enough, the absolute error term will be very small, such that the bounds become essentially relative. We moreover argued that this absolute term is realistic for eigenvalues computed on real computers using finite precision floating point architectures. Thus, in a certain sense, we achieved the goal of describing the observed behavior of the eigenvalues in two different ways: we were able to prove that the approximation errors scale with the size of the true eigenvalues and that they will stagnate at a certain (very small) level.

Concerning the spectral projections, there exists a similar numerically observable effect which lacked a matching counterpart in theory. Scalar products with eigenvectors of small eigenvalues seemed to fluctuate much less than those with eigenvectors of large eigenvalues. Here, we did not provide an independent convergence proof of our own but rather complemented an existing result with a relative-absolute envelope which again scales with the eigenvalues. We proved that the scalar products also show a nice convergence behavior: Scalar products with eigenvectors of small eigenvalues which are very small asymptotically are already very small for finite sample sizes. This result in itself seems rather abstract, but has proven to have powerful consequences (see below).

Principal component analysis directly suggests itself as a field of application for these results, mostly due to the fact that principal component analysis consists of the computation of the

eigenvalues of a symmetric matrix, and in the case of kernel PCA, even the eigenvalues of the kernel matrix itself. We were able to readily derive three results covering almost all aspects of the convergence of the estimated principal values. It should be stressed that these results are not dependent on strong assumptions on the underlying probability measure. The only requirement is that the true eigenvalues decay quickly. This constraint is usually fulfilled for smooth kernel functions. First of all, we derived a purely multiplicative bound for the principal values in a finite-dimensional setting. A strength of the result lies in the fact that the convergence speed is expressed with respect to the norm of certain error matrices. Very generally, the convergence can be shown to depend on the fourth moment of the underlying measure along the principal directions. However, if additional knowledge on the distribution is available, one might be able to provide a detailed analysis of the size of the error matrix, which can then result in much faster convergence results. Put differently, the convergence speed does not simply depend on a single parameter of the probability distribution, but on a complex object which can be studied further for the cases one is interested in.

The second and third result treat kernel PCA, a non-linear extension of principal component analysis. Using the relative-absolute bounds for the eigenvalues, we showed that kernel PCA approximates the true principal components with high precision. For kernel PCA, an interesting question is that of the effective dimension. Since the principal values usually have no special structure besides decaying at some rate, one often projects to a number of leading dimensions such that the reconstruction error becomes small enough. This error is linked to the sums of all eigenvalues except for the first few. This reconstruction error has been a natural target for the approach to prove convergence of the eigenvalues by the Courant-Fisher characterization. Using the relative-absolute perturbation bound on the eigenvalues of the kernel matrix, we were able to prove a relative-absolute bound on the reconstruction error, which scales nicely as the eigenvalues decay rapidly. This result is a significant improvement compared with previous results which did not scale with the size of the eigenvalues involved.

An interesting consequence of the result on the reconstruction error is that a finite sample in feature space will always be contained in a low-dimensional subspace of the (possibly infinite-dimensional) feature space. This number does not depend on the number of samples, but rather becomes even more stable as the number of sample grows. Therefore, the general intuition that learning in feature space is hard because the data fully covers an n dimensional subspace spanned by the data points is wrong. Indeed, while it is true that the data spans an n dimensional subspace, only a few directions have large variance. The rôle of regularization then becomes that of adjusting the scale at which the algorithms works, such that the algorithm only sees the finite-dimensional part of the data.

In the context of supervised learning, we first studied the relation between the label information and the kernel matrix in an algorithm independent fashion. The assumption is that the target function can be represented in terms of the kernel matrix. The crucial point here was the transformation of the label vector to its representation with respect to the eigenbasis of the kernel matrix. Then, it follows by the results on spectral projections that the information content of the label, in the case of regression given as the smooth target function, is contained in the first few coefficients (when coefficients are ordered with respect to non-increasing eigenvalues), while the noise is evenly distributed over all of the coefficients.

The essential finite-dimensionality of the object samples and the label vector combined can be seen as a more direct version of the well known fact that at a non-zero scale, the set of all hypotheses with bounded weight vector in an infinite-dimensional space has finite VC-dimension (Evgeniou and Pontil, 1999; Alon et al., 1997).

This picture has some resemblance with that of performing a Fourier analysis of a signal with additive noise. There, the signal is also contained in some frequency band, while the noise covers all of the spectrum. The strength of the kernel approach then lies in the fact that this decomposition can be carried out over arbitrary spaces on which smooth kernels can be defined, and for all geometries of sample points. Fourier analysis is usually confined to compact rectangular domains in low dimensions.

The structure of the label vector with respect to the eigenbasis of the kernel function suggests

the definition of a cut-off dimensions d which we defined as the number such that the information content of the label vector is completely contained in the first d coefficients. We showed that these cut-off dimensions can be effectively estimated by proposing two different procedures and testing them extensively on different data sets. The approach based on performing a maximum likelihood fit with a two component model proved to be the more robust and reliable variant.

We also discussed using the cut-off dimension estimators to perform a structural analysis of a given data set, again in an algorithm independent fashion. It turns out that combining the cut-off dimension estimators with a family of kernels depending on a scale parameter, one can detect structure at different scales by estimating cut-off dimensions at varying kernel widths.

Finally, we have turned to kernel ridge regression as an example of a supervised kernel method. The advantage of kernel ridge regression is that the training step is computed by a matrix which is closely related to the kernel matrix. Based on our knowledge of the spectral structure of the kernel matrix, the training step can be fully decomposed and analyzed. We have seen that kernel ridge regression basically amounts to low-pass filtering of the signal. Again, the advantage with respect to employing a Fourier decomposition is that kernel ridge regression can be painlessly extended to kernels in arbitrary dimensions. Furthermore, the basis functions in kernel ridge regression adapt themselves to the underlying probability density.

The free parameter of kernel ridge regression is the regularization constant. Based on the analysis, it seems that this regularization constant should be chosen according to the cut-off dimension. The resulting method was called the *spectrum method*. During extensive experiments both for regression and classification we were able to show that the spectrum method performed very competitively with existing state-of-the-art methods. While we have to admit that there is really no shortage of good model selection methods, these results show that the theoretical analysis and the insights into kernel ridge regression were actually sufficiently relevant to allow us to propose a competitive method for model selection which uses only the structural insights into the spectrum of the label vector to perform effective model selection.

Future Directions

We believe that some of the results have interesting theoretical implications which we have only briefly touched upon.

We have shown that both the object samples as well as the label vector have an essentially finite dimensional structure at a given scale with the dimension not depending on the sample size. The question is if this characterization can be used to explain in a more direct fashion, without involving VC-dimension arguments, why learning in feature spaces work well?

Closely linked to this question is if the effective dimension in feature space and the cut-off dimension of the label vector can be used as some form of *a priori complexity measure* for data sources. An existing problem with data-dependent error estimates lies in the fact that the dependency on the data is often constructed in such a way that the complexity of the data set only becomes apparent after the learning has taken place, for example by realizing a certain margin. A problem of this kind of argument has the drawback that one cannot ensure a priori that an algorithm performs well. On the other hand, the effective dimension of the data set and the cut-off dimension of the label depend only on the chosen kernel which is a large step towards an a priori complexity measure. In particular, because the size of these dimensions is already proven to converge, it is even possible to effectively estimate these quantities.

Of course, this question ultimately has to lead to generalization error bounds for kernel ridge regression. The question thus is, given that the cut-off dimension of the data is known, can we bound the generalization error for kernel ridge regression? In principle, we can already estimate the size of the in-sample error between the fitted function and the target function. In order to bound the out-of-sample error, one has to consider how well the Nyström extrapolates of the eigenvectors predict. These could be handled using estimates on their Lipschitz constants. This way, one could derive an estimate of the generalization error which is directly linked to how the algorithm works, in contrast to using some abstract capacity argument based on VC-theory. This approach could have the added benefit of obtaining a better intuitive understanding of how the

algorithm works based on the theoretical analysis, in contrast to capacity arguments which tend to consider the algorithm as a black box which simply selects some solution from a hypothesis set in a non-transparent fashion.

In my opinion, it proved possible and rewarding to perform detailed analyses of specific algorithms and objects. In the best case, this can be both interesting and relevant. I'd like to close this thesis with the following sentence which I borrowed from the end of Bauer (1990).

On ne finit pas un œuvre,
on l'abandonne.

(Gustave Flaubert)

Bibliography

754-1985, I. S. (1985). IEEE Standard for Binary Floating-Point Arithmetic. IEEE Computer Society.

Abramowitz, M. and Stegun, I. A., editors (1972). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, 9th printing*, chapter 22, "Legendre Functions", and chapter 8, "Orthogonal Polynomials", pages 331–339, 771–802. Dover, New York.

Ahrendt, T. (1999). *Schnelle Berechnung der Exponentialfunktion auf hohe Genauigkeit*. PhD thesis, Mathematisch-Naturwissenschaftliche Fakultät, Universität Bonn. ("Fast Computation of the Exponential Funktion to High Precision", in German).

Alon, N., Ben-David, S., Cesa-Bianchi, N., and Haussler, D. (1997). Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM*, 44(4):615–631.

Anderson, T. W. (1963). Asymptotic theory for principal component analysis. *Annals of Mathematical Statistics*, 34:122–148.

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis*. Wiley-Interscience, 3rd edition.

Anselone, P. M. (1971). *Collectively Compact Operator Approximation Theory and Applications to Integral Equations*. Prentice-Hall Series in Automatic Computation. Prentice-Hall, Englewood Cliffs, New Jersey.

Atkinson, K. E. (1997). *The Numerical Solution of Integral Equations of the Second Kind*. Cambridge University Press.

Baker, C. T. H. (1977). *The numerical treatment of integral equations*. Clarendon Press, Oxford.

Bauer, H. (1990). *Wahrscheinlichkeitstheorie*. de Gruyter Lehrbuch. de Gruyter, Berlin, New York, 4th edition. ("Probability theory", in German).

Bengio, Y., Delalleau, O., Le Roux, N., Paiement, J.-F., Vincent, P., and Ouimet, M. (2004). Learning eigenfunctions links spectral embedding and kernel PCA. *Neural Computation*, 16:2197–2219.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

Chatterjee, C., Roychowdhury, V. P., and Chong, E. K. P. (1998). On relative convergence properties of principal component analysis algorithms. *IEEE Transactions on Neural Networks*, 9(2).

Cover, T. M. (1969). Learning in pattern recognition. In Watanabe, S., editor, *Methodologies of Pattern Recognition*, pages 111–132, New York. Academic Press.

Cristianini, N. and Shawe-Taylor, J. (2000). *Support Vector Machines and other kernel-based learning methods*. Cambridge University Press.

Dauxois, J., Pousse, A., and Romain, Y. (1982). Asymptotic theory for the principal component analysis of a vector random function: Some applications to statistical inference. *Journal of Multivariate Analysis*, 12:136–154.

Davis, C. and Kahan, W. M. (1970). The rotation of eigenvectors by a perturbation, iii. *SIAM Journal of Numerical Analysis*, 7:1–46.

Devroye, L., Györfi, L., and Lugosi, G. (1996). *A Probabilstic Theory of Pattern Recognition*. Springer Verlag.

Donoho, D. L. and Johnstone, I. M. (1995). Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90:1200–1224.

Donoho, D. L. and Johnstone, I. M. (1998). Minimax estimation via wavelet shrinkage. *The Annals of Statistics*, 26(3):879–921.

Donoho, D. L., Johnstone, I. M., Kerkyacharian, G., and Picard, D. (1995). Wavelet shrinkage: Asymptopia? *Journal of the Royal Statistical Society*, 57:301–369.

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. John Wiley & Sons, 2nd edition.

Eisenstat, S. C. and Ipsen, I. C. F. (1994). Relative perturbation bounds for eigenspaces and singular vector subspaces. In *5th SIAM Conference on Applied Linear Algebra*, pages 62–65, Philadelphia. SIAM.

Engl, H. W. (1997). *Integralgleichungen*. Springer-Verlag. ("Integral equations", in German).

Evgeniou, T. and Pontil, M. (1999). On the V_γ dimension for regression in reproducing kernel Hilbert spaces. In *Proceedings of Algorithmic Learning Theory*, Tokyo, Japan.

Girosi, F., Jones, M., and Poggio, T. (1995). Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269.

Goldberg, P. W., Williams, C. K. I., and Bishop, C. M. (1998). Regression with input-dependent noise: A gaussian process treatment. In Jordan, M. I., Kearns, M. J., and Solla., S. A., editors, *Advances in Neural Information Processing Systems*, volume 10. Lawrence Erlbaum.

Golub, G. H. and van Loan, C. F. (1996). *Matrix Computations*. Johns Hopkins University Press, 3rd edition.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference and Prediciton*. Springer-Verlag.

Herbrich, R. (2002). *Learning Kernel Classifiers*. MIT Press.

Hoeffding, W. (1963). Probability inequalities for sums of bounded variables. *Journal of the American Statistical Association*, 58:13–30.

Hoffman, A. and Wielandt, H. (1953). The variation of the spectrum of a normal matrix. *Duke. Math. J.*, 29:37–38.

Horn, R. A. and Johnson, C. R. (1985). *Matrix Analysis*. Cambridge University Press.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer-Verlag New York Inc., 2nd edition.

Kato, T. (1976). *Perturbation Theory for Linear Operators*. Springer-Verlag Berlin, 2nd edition.

Koltchinskii, V. and Giné, E. (2000). Random matrix approximation of spectra of integral operators. *Bernoulli*, 6(1):113–167.

Koltchinskii, V. I. (1998). Asymptotics of spectral projections of some random matrices approximating integral operators. *Progress in Probability*, 43:191–227.

Kotel'nikov, V. A. (1933). On carrying capacity of "ether" and wire in electro-communications. *Material for the First All-Union Conference on Questions of Communications. Izd. Red. Upr. Svyazi RKKA (Moscow)*. (in Russian).

Lepskii, O. V. (1990). On one problem of adaptive estimation on white gaussian noise. *Theory of Probability and its Applications*, 35:454–466.

McDiarmid, C. (1989). On the method of bounded differences. In *Surveys in Combinatorics 1989*, pages 148–188. Cambridge University Press.

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

Nyström, E. J. (1930). Über die praktische Auflösung von Integralgleichungen mit Anwendung auf Randwertaufgaben. *Acta Mathematica*, 54:185–204.

Pestman, W. R. (1998). *Mathematical Statistics*. Willem de Gruyter, Berlin.

Rätsch, G., Onoda, T., and Müller, K.-R. (2001). Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320. also NeuroCOLT Technical Report NC-TR-1998-021.

Rifkin, R. (2002). *Everything Old Is New Again: A Fresh Look at Historical Approaches in Machine Learning*. PhD thesis, Massachusetts Institute of Technology.

Schmidt, R. O. (1986). Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, AP-34(3).

Schölkopf, B. (1997). *Support Vector Learning*. PhD thesis, Technische Universität Berlin.

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

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

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels*. MIT Press.

Shannon, C. (1949). Communication in the presence of noise. *Proceedings of the Institute of Radio Engineers*, pages 10–21.

Shawe-Taylor, J., Cristianini, N., and Kandola, J. (2002a). On the concentration of spectral properties. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA. MIT Press.

Shawe-Taylor, J., Williams, C., Cristianini, N., and Kandola, J. (2002b). On the eigenspectrum of the gram matrix and its relationship to the operator eigenspectrum. In N. Cesa-Bianchi et al., editor, *ALT 2002*, volume 2533 of *Lecture Notes in Artifical Intelligence*, pages 23–40. Springer-Verlag Berlin Heidelberg.

Shawe-Taylor, J. and Williams, C. K. I. (2003). The stability of kernel principal components analysis and its relation to the process eigenspectrum. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems*, volume 15.

Shawe-Taylor, J., Williams, C. K. I., Cristianini, N., and Kandola, J. (2004). On the eigenspectrum of the gram matrix and the generalisation error of kernel PCA. Technical Report NC2-TR-2003-143, Department of Computer Science, Royal Holloway, University of London. Available from `http://www.neurocolt.com/archive.html`.

Steele, J. M. (1997). *Probability Theory and Combinatorial Optimization*. SIAM, Philadelphia.

Stein, C. (1981). Estimation of the mean of a multivariate normal distribution. *Annals of Statistics*, 9(6):1040–1053.

Stewart, G. and Sun, J. (1990). *Matrix Perturbation Theory*. Academic Press, New York.

van der Laan, M. J., Dudoit, S., and Keles, S. (2004). Asymptotic optimality of likelihood-based cross-validation. *Statistical Applications in Genetics and Molecular Biology*, 3(1). Article 4.

van der Vaart, A. W. and Wellner, J. A. (1998). *Weak Convergence and Empirical Processes*. Springer-Verlag.

Vapnik, V. (1998). *Statistical Learning Theory*. John Wiley & Sons.

von Luxburg, U. (2004). *Statistical Learning with Similarity and Dissimilarity Functions*. PhD thesis, Technische Universität Berlin.

Wahba, G. (1990). *Spline Models For Observational Data*. Society for Industrial and Applied Mathematics.

Weyl, H. (1968). *Gesammelte Abhandlungen*. Springer, Berlin.

Williams, C. K. I. and Rasmussen, C. E. (1996). Gaussian processes for regression. In Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E., editors, *Advances in Neural Information Processing Systems*, volume 8. MIT Press.

Williams, C. K. I. and Seeger, M. (2000). The effect of the input density distribution on kernel-based classifiers. In Langley, P., editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*. Morgan Kaufmann.

Williamson, R. C., Smola, A., and Schölkopf, B. (2001). Generalization performance of regularization networks and support vector machines via entropy numbers of compact operators. *IEEE Transactions on Information Theory*, 47(6):2516–2532.

Zwald, L., Bousquet, O., and Blanchart, G. (2004). Statistical properties of kernel principal component analysis. In Shawe-Taylor, J. and Singer, Y., editors, *Learning Theory: 17th Annual Conference on Learning Theory, COLT 2004.*, volume 3120/2004 of *Lecture Notes in Computer Science*, pages 594–608. Springer-Verlag Heidelberg.

Zwald, L., Vert, R., Blanchard, G., and Massart, P. (2005). Kernel projection machine: a new tool for pattern recognition. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA.

Danksagungen

Die in dieser Dissertation destillierte Forschungsarbeit stellt das Ergebnis eines Unternehmens dar, das einige Jahre meines Lebens eingenommen hat. Wie es in der Natur eines solchen Vorhabens liegt, hat es dabei einige Höhen und Tiefen gegeben. Während dieser Zeit habe ich immer auf die Unterstützung von Verwandten, angeheirateten Verwandten, Freunden und Kollegen zählen können, für die ich mich an dieser Stelle im Einzelnen bedanken möchte.

Ganz besonders hervorheben möchte ich hierbei meine Frau Katrin, die mich durch den Prozeß begleitet hat und besonders in der Endphase die eine oder andere Entbehrung auf sich nehmen musste. Ebenso möchte ich meinen Eltern für ihr tiefes Vertrauen und ihre Unterstützung danken.

Ich möchte Professor Joachim Buhmann für die langjährige zuverlässige Unterstützung und die Freiheit bei der Entfaltung dieses interessanten Forschungsvorhabens, die mir großzügig eingeräumt wurde, danken. Außerdem möchte ich Professor Michael Clausen für die Übernahme des Zweitgutachtens, sowie den Professoren Rolf Klein und Sergio Albeverio für Ihre Zeit danken, die sie mir mit der Teilnahme an der Prüfungskommission opfern.

Die schließliche Fertigstellung dieser Arbeit hing wesentlich von der Unterstützung und Toleranz Professor Klaus-Robert Müller ab, in dessen Gruppe ich kurz vor der Fertigstellung der Dissertation gewechselt war.

Meine alten Freunde aus Bonn, insbesondere Tilman Lange, Daniel Hanisch, Klaus Ostermann, Florian Sohler, Axel Mosig, Gero Müller, Peter Orbanz, Frank Ciesinski, und Dominik Schrader gebührt der Dank für all die schönen Jahre, die gute Zusammenarbeit und die moralische Unterstützung. Dasselbe gilt für meine alten Kollegen aus Bonn, Volker Roth, Thomas Zöller, Lothar Hermes, Bernd Fischer, Weijun Chen, Thorsten Belker, Dirk Schulz, Jürgen Schumacher und Mark Moors.

Und schließlich gilt mein Dank noch: Meiner Schwägerin Miriam für die Beschaffung des Buches von Anselone, das mir sehr weitergeholfen hat. Axel Mosig möchte ich neben vielem anderen für die Durchsicht einiger Kapitel und für die mentale Unterstützung danken. Peter Orbanz für die Beschaffung einiger Paper, für die moralische Unterstützung und für einige Korrekturvorschläge. Christin Schäfer für's Korrekturlesen. Stefan Harmeling, Gilles Blanchard und Motoaki Kawanabe für hilfreiche Diskussionen.

Lebenslauf

Persönliche Daten:

Name, Vorname:	Braun, Mikio Ludwig
Geburtsdatum, -ort:	7. April 1975, Brühl (Rheinland)
Anschrift:	Lebuser Str. 14, 10243 Berlin
Telefon (privat):	030/42105642
E-Mail:	mikio@first.fhg.de

Ausbildung:

1981 – 1985	Gesamtgrundschule West in Brühl
1985 – 1989	Max-Ernst-Gymnasium in Brühl
1989 – 1994	Apostelgymnasium in Köln Abschluß: Abitur (Note: 1.1)
August 1994 – Oktober 1995	Zivildienst beim Malteser Hilfsdienst in Köln
Oktober 1995 - Oktober 2000	Studium der Informatik an der Universität Bonn Abschluß: Diplom (mit Auszeichnung)
Oktober 1996 – Februar 2003	Studium der Mathematik an der Universität Bonn Abschluß: Vordiplom (Note: 1.0)
Oktober 2000 – Juli 2005	Promotionsstudium Informatik