

On Relevant Dimensions in Kernel Feature Spaces

Mikio L. Braun

*Technische Universität Berlin
Franklinstr. 28/29, FR 6-9
10587 Berlin, Germany*

MIKIO@CS.TU-BERLIN.DE

Joachim M. Buhmann

*Institute of Computational Science
ETH Zurich, Universitätstrasse 6
CH-8092 Zürich, Switzerland*

JBUEHMANN@INF.ETHZ.CH

Klaus-Robert Müller*

*Technische Universität Berlin
Franklinstr. 28/29, FR 6-9
10587 Berlin, Germany*

KRM@CS.TU-BERLIN.DE

Editor: Peter Bartlett

Abstract

We show that the relevant information of a supervised learning problem is contained up to negligible error in a finite number of leading kernel PCA components if the kernel matches the underlying learning problem in the sense that it can asymptotically represent the function to be learned and is sufficiently smooth. Thus, kernels do not only transform data sets such that good generalization can be achieved using only linear discriminant functions, but this transformation is also performed in a manner which makes economical use of feature space dimensions. In the best case, kernels provide efficient implicit representations of the data for supervised learning problems. Practically, we propose an algorithm which enables us to recover the number of leading kernel PCA components relevant for good classification. Our algorithm can therefore be applied (1) to analyze the interplay of data set and kernel in a geometric fashion, (2) to aid in model selection, and (3) to denoise in feature space in order to yield better classification results.

Keywords: kernel methods, feature space, dimension reduction, effective dimensionality

1. Introduction

Kernel machines implicitly map the data into a high-dimensional feature space in a non-linear fashion using a kernel function. This mapping is often referred to as an *empirical kernel map* (Schölkopf et al., 1999; Vapnik, 1998; Müller et al., 2001; Schölkopf and Smola, 2002). By virtue of the empirical kernel map, the data is ideally transformed in a way such that a linear discriminative function can separate the classes with low generalization error by a canonical hyperplane with large margin. Such large margin hyperplanes provide an appropriate mechanism of capacity control and thus “protect” against the high dimensionality of the feature space.

However, this picture is incomplete as it does not explain why the typical variants of capacity control cooperate well with the induced feature map. This paper adds a novel aspect as the key idea

*. Also at Fraunhofer FIRST.IDA, Kekuléstr. 7, 12489 Berlin, Germany.

to this picture. We show theoretically that if the learning problem matches the kernel well, the relevant information of a supervised learning data set is always contained in the subspace spanned by a finite and typically small number of leading kernel PCA components (principal component analysis in the feature space induced by the kernel, see below and Section 2), up to negligible error. This result is based on recent approximation bounds for the eigenvectors of the kernel matrix which show that if a function can be reconstructed using only a few kernel PCA components asymptotically, then the same already holds in a finite sample setting, even for small sample sizes.

Consequently, the use of a kernel function not only greatly increases the expressive power of linear methods by non-linearly transforming the data, but it does so ensuring that the high dimensionality of the feature space does not become overwhelming: the relevant information for learning stays confined within a comparably *low*-dimensional subspace. This finding underlines the efficient use of data that is made by kernel machines if the kernel works well for the learning problem. A smart choice of kernel permits to make better use of the available data at a favorable “number of data points per effective dimension”-ratio, even for infinite-dimensional feature spaces. The kernel induces an efficient representation of the data in feature space such that even unregularized methods like linear least squares regression are able to perform well on the reduced feature space.

Let us consider an example. Figure 1(a) shows a two-dimensional classification problem (the *banana* data set from Rätsch et al., 2001). We can visualize the contributions of the individual kernel PCA components¹ to the class membership by plotting the absolute values of scalar products between the labels and the kernel PCA components. Figure 1(b) shows the resulting contributions sorted by decreasing principal value (variance along principal direction). We can observe that the contributions are concentrated in the leading kernel PCA directions, but a large fraction of the information is contained in the later components as well.

Note, however, that the class membership information in the data set also contains a certain amount of noise. Therefore, Figure 1(b) actually shows a mixture of relevant information and noise. We need to devise a different procedure for assessing the amount of task-relevant information in certain kernel PCA components. This can be accomplished by incorporating a second data set from the same source for testing. One first projects onto the subspace spanned by a number of leading kernel PCA components, trains a linear classifier (for example, by least squares regression) and then measures the prediction error on the test set. The test error is large either if the considered subspace did not capture all of the relevant information, or if it already contained too much noise leading to overfitting. If the minimal test error is on par with a state-of-the-art method independently trained using the same kernel then the subspace has successfully captured all of the relevant information.

If we apply this procedure to our data set, we obtain training and test errors as shown in Figure 1(c). By definition, the training error decreases as more and more dimensions are used. However, after decreasing quickly initially, the test error eventually starts to increase again. The minimal test error also coincides with the actually achievable test error using, for example, support vector machines. Therefore, we see that the later components only contain noise, and the relevant information is contained in the leading kernel PCA components. In this paper, our goal is to understand

1. Recall that kernel PCA (Schölkopf et al., 1998) amounts to implicitly performing PCA in the feature space. Roughly, instead of the covariance matrix, one considers the eigenvalues and eigenvectors of the kernel matrix, which is built from all pairwise evaluations of the kernel matrix on the inputs. Principal values (variances) are still given by the eigenvalues of the kernel matrix, but principal directions (which would be potentially infinite-dimensional vectors) are replaced by principal components, which are scalar products with the principal directions. Also see Section 2.

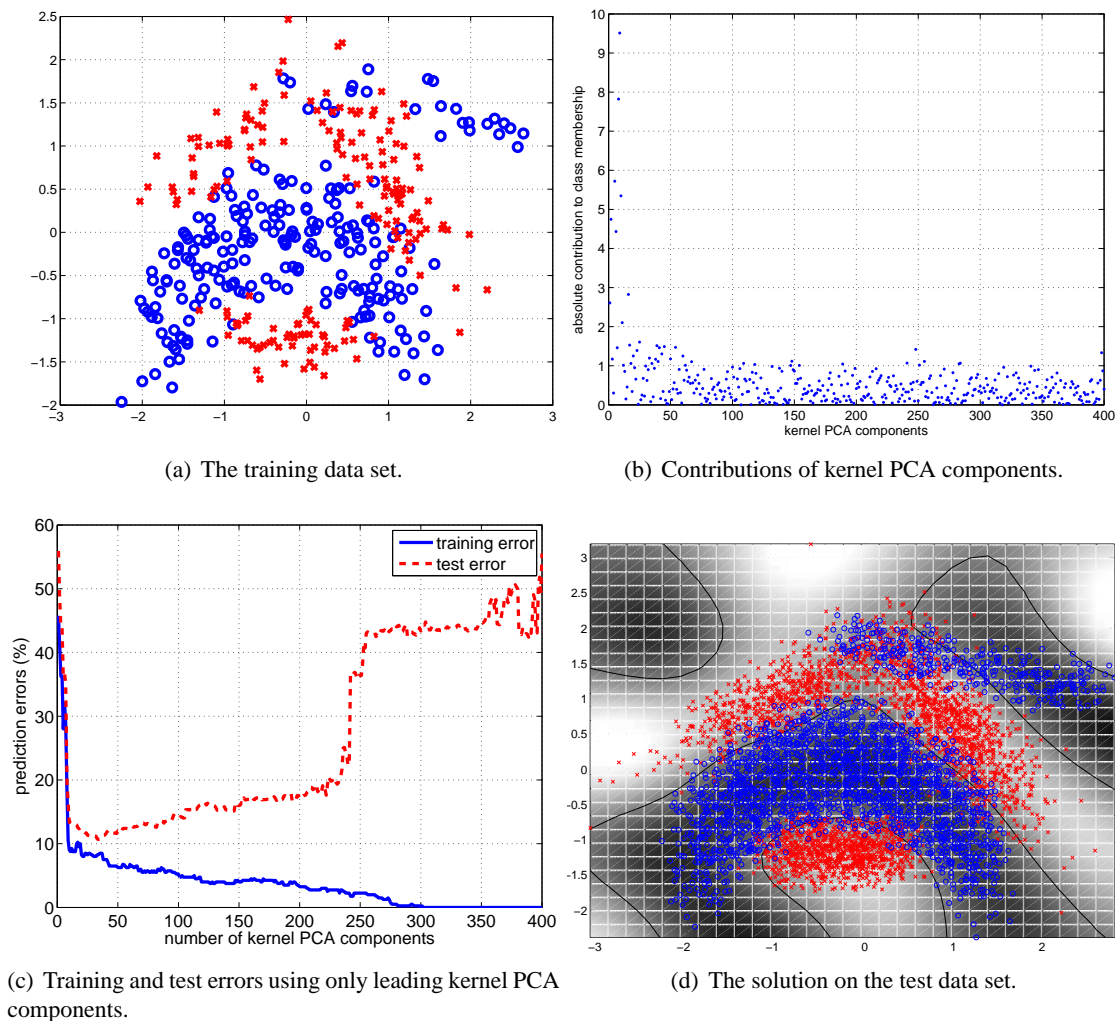


Figure 1: A more complex example (resample 1 of the “banana” data set, see Section A). This time, the information is not contained in a single component. Nevertheless, the test error of a hyperplane learned using only the first d components has a clear minimum at $d = 34$ at optimal error rate (cf. Table 3), showing that the relevant information is contained in the leading 34 directions.

more thoroughly why and when this effect occurs, and to estimate the dimensionality of a concrete data set given a kernel.

Our claim—that the relevant information about a learning problem is contained in the space spanned by the leading kernel PCA components—is similar to the idea that the information about the learning problem is contained in the kernel PCA components with the largest contributions. However, our results show that the magnitude of the contribution of a kernel PCA component to the label information is only partially indicative of the relevance of that component. Instead, we show that the leading kernel PCA components (sorted by corresponding principal value) contain

the relevant information. Components which contain only little variance will therefore not contain relevant information. If such a component manages to contribute much to the label information, it will only reflect noise.

What practical implications follow from these results? We explore several possibilities of using these ideas to assess the suitability of a kernel or a family of kernels to a specific data set. The main idea is that the observed dimensionality of the data set in feature space is characteristic for the relation between a data set and a kernel. Roughly speaking, the relevant dimensionality of the data set corresponds to the complexity of the learning problem when viewed through the “lens” of the kernel function. Using the estimated dimensionality, one can project the labels onto the corresponding subspace and obtain a noise free version of the labels. By comparing the denoised labels to the original labels, one can estimate of the amount of noise contained in the labels. One therefore obtains a more detailed measure of the fit between the kernel and the data set as compared to, for example, the cross-validation error alone. This allows us to take a closer look at data sets on which the achieved error is quite large. In such cases, we are able to distinguish whether the data set is highly complex and the amount of data is insufficient, or the amount of intrinsic noise is very large. This is practically relevant as one has to deal with both these cases quite differently, either by providing more data, or by thinking about means to obtain less noisy or ambiguous features.

We summarize the main contributions of this paper: (1) We provide theoretical bounds showing that the relevant information (defined in Section 2) is actually contained in the leading projected kernel principal components under appropriate conditions. (2) We propose an algorithm which estimates the relevant dimensionality and related estimates of the data set and permits to analyze the appropriateness of a kernel for the data set, and thus to perform model selection among different kernels. (3) We validate the accuracy of the estimates experimentally by showing that non-regularized methods perform on the reduced feature space on par with state-of-the-art kernel methods. We analyze some well-known benchmark data sets in Section 5. Note that we do not claim to obtain better performance within our framework when compared to, for example, cross-validation techniques. Rather, we are on par. Our contribution is to foster an understanding about a data set and to gain better insights of whether a mediocre classification result is due to intrinsic high dimensionality of the data (and consequently insufficient number of examples), or an overwhelming noise level.

2. Preliminaries

Let us start to formalize the ideas introduced so far. As usual, we consider a data set $(X_1, Y_1), \dots, (X_n, Y_n)$ where the inputs X lie in some space \mathcal{X} and the outputs Y to be predicted are in $\mathcal{Y} = \{\pm 1\}$ for classification or $\mathcal{Y} = \mathbb{R}$ for regression. We often refer to the outputs Y_i as the “labels” irrespective of whether we are considering a classification or regression task. We assume that the (X_i, Y_i) are drawn i.i.d. from some probability measure $\mathbb{P}_{\mathcal{X} \times \mathcal{Y}}$. In kernel methods, the data is non-linearly mapped into some feature space \mathcal{F} via the feature map Φ . Scalar products in \mathcal{F} can be computed by the kernel k in closed form: $\langle \Phi(x), \Phi(x') \rangle = k(x, x')$. Summarizing all the pairwise scalar products results in the (normalized) kernel matrix \mathbf{K} with entries $k(X_i, X_j)/n$.

In the discussion below, we study the relationship between the *label vector* $Y = (Y_1, \dots, Y_n)$ and the kernel PCA components which are introduced next. Kernel PCA (Schölkopf et al., 1998) is a kernelized version of PCA. Since the dimensionality of the feature space might be too large to deal

Symbol	Meaning
n	number of training examples
$X_i \in \mathcal{X}$	input examples
$Y_i \in \mathcal{Y}$	output labels
$k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$	kernel function
$\mathbf{K} = (k(X_i, X_j))/n$	(normalized) kernel matrix
$Y = (Y_1, \dots, Y_n)$	label vector
$\Phi: \mathcal{X} \rightarrow \mathcal{F}$	feature map
$l_m \in \mathbb{R}_{\geq 0}$	m th kernel PCA value (in descending order), m th eigenvalue of kernel matrix \mathbf{K}
$v_m \in \mathcal{F}$	m th kernel PCA direction
$f_m(x) = \langle \Phi(x), v_m \rangle$	m th kernel PCA component
$u_m = (f_m(X_1), \dots, f_m(X_n))$	m th kernel PCA component evaluated on X_1, \dots, X_n , m th eigenvector of kernel matrix \mathbf{K}
$\pi_d(Y) = \sum_{i=1}^d u_i u_i^\top Y$	projection onto first d kernel PCA components
$G = (E(Y_1 X_1), \dots, E(Y_n X_n))$	relevant information vector
$z_i = u_i^\top G$	contribution of i th eigenvector to relevant information
$g(x) = E(Y X = x)$	relevant information function
$\mathcal{L}^2(\mathcal{X}, P_X)$	set of all square integrable functions with respect to P_X
$T_k f(s) = \int_{\mathcal{X}} k(s, t) f(t) P_X(dt)$	integral operator associated with k
$\lambda_i \in \mathbb{R}_{\geq 0}$	i th eigenvalue of T_k
$\psi_i \in \mathcal{L}^2(\mathcal{X}, P_X)$	i th eigenfunction of T_k
$\zeta_i = \langle \psi_i, g \rangle$	contribution of i th eigenfunction to relevant information
\hat{d}	estimated relevant dimension
cv_{100}	leave-one-out cross-validation error
\hat{G}	estimated relevant information vector
$\mathbf{S} = \sum_{i=1}^d u_i u_i^\top$	“hat”-matrix
$\hat{\epsilon}$	estimated noise-level

Table 1: Overview of notation used in this paper.

with the vectors directly, the principal directions are represented using the points X_i of the data set:

$$v_m = \sum_{i=1}^n \alpha_i \Phi(X_i),$$

where $\alpha_i = [u_m]_i / l_m$, $[u_m]_i$ is the i th component of the m th eigenvector of the kernel matrix \mathbf{K} , and l_m the corresponding eigenvalue.² Still, v_m can usually not be computed explicitly such that one instead works with *kernel PCA components*

$$f_m(x) = \langle \Phi(x), v_m \rangle.$$

We are interested in the relation between f_m and a label vector Y . As we have seen in the introduction, it seems that only a finite number of leading kernel PCA components are necessary to represent the relevant information about the learning problem up to a small error.

2. As usual, we assume that l_m and u_m have been sorted such that $l_1 \geq \dots \geq l_n$.

Therefore, we would like to compare f_m with the values Y_1, \dots, Y_n at the points X_1, \dots, X_n . The following easy lemma summarizes the relationship between the sample vector of f_m and Y .

Lemma 1 *The m th kernel PCA component f_m evaluated on the X_i s is equal to the m th eigenvector of the kernel matrix \mathbf{K} : $(f_m(X_1), \dots, f_m(X_n)) = u_m$. Consequently, the sample vectors are orthogonal, and the projection of a vector $Y \in \mathbb{R}^n$ onto the leading d kernel PCA components is given by $\pi_d(Y) = \sum_{m=1}^d u_m u_m^\top Y$.*

Proof The m th kernel PCA component for a point X_j in the training set is

$$f_m(X_j) = \langle \Phi(X_j), v_m \rangle = \frac{1}{l_m} \sum_{i=1}^n \langle \Phi(X_j), \Phi(X_i) \rangle [u_m]_i = \frac{1}{l_m} \sum_{i=1}^n k(X_j, X_i) [u_m]_i.$$

The sum computes the j th component of $\mathbf{K}u_m$, and $\mathbf{K}u_m = l_m u_m$, because u_m is an eigenvector of \mathbf{K} . Therefore

$$f_m(X_j) = \frac{1}{l_m} [l_m u_m]_j = [u_m]_j.$$

Since \mathbf{K} is a symmetric matrix, its eigenvectors u_m are orthonormal, and the projection of Y onto the space spanned by the first d kernel PCA components is given by $\sum_{m=1}^d u_m u_m^\top Y$. ■

Since the kernel PCA components are orthogonal, the coefficients of a vector $Y \in \mathbb{R}^n$ with respect to the basis u_1, \dots, u_n is easily computed by forming the scalar products. We call the coefficients

$$z_m = u_m^\top Y \tag{1}$$

of Y w.r.t. the basis formed from the kernel PCA components the *kernel PCA coefficients*. They are the central object of our discussion.

The projection of Y to a kernel PCA component can be thought of as the least squares regression of Y using only the direction along the kernel PCA component in feature space.

Using the kernel PCA coefficients, we can extend the projected labels to new points via

$$\hat{Y}(x) = \sum_{m=1}^d z_m f_m(x),$$

which amounts to the prediction of least squares regression on the reduced feature space.

3. The Label Vector and Kernel PCA Components

In the introduction, we have discussed an example which suggests that a small number of leading kernel PCA components might suffice to capture the relevant information about the output variable. It is clear that we cannot expect this behavior for all possible data sets and kernels. It seems plausible though, that under certain conditions, the distribution of the data and the kernel fit together well. Then we can expect to observe this behavior with high probability for a random sample from this distribution through some form of concentration or convergence property.

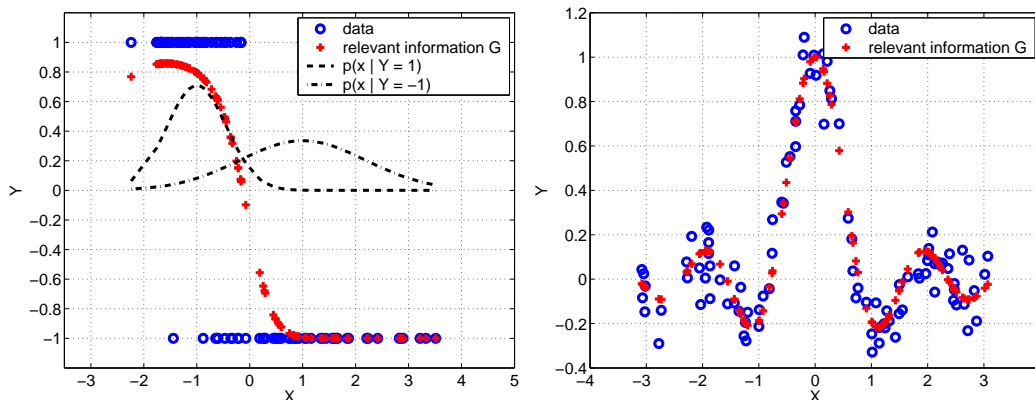


Figure 2: Relevant information vectors visualized for the classification and the regression case. In the (two-class) classification case (left) it encodes the posterior probability (scaled between -1 and 1), in the regression case it is the sample vector of the function to be learned.

3.1 Decomposing the Label Vector Information

We start the discussion by defining formally what the relevant information contained in the labels is. Given a label vector Y , we define the relevant information vector as the vector of the expected labels:

$$G = (E(Y_1|X_1), \dots, E(Y_n|X_n)).$$

Intuitively speaking, G is a noise-free version of Y . This vector contains all the relevant information about the outputs Y of the learning problem: For regression, G amounts to the values of the true function. For the case of two-class classification, the vector G contains all the information about the optimal decision boundary. Since $E(Y|X) = P(Y = 1|X) - P(Y = -1|X)$, the sign of G contains the relevant information on the true class membership by telling us which class is more probable (see Figure 2 for examples). Thus, using this denoised label information, the learning problem becomes much easier as the denoised labels already contain the Bayes optimal prediction at that point.³

Using G we obtain a very useful additive decomposition of the labels into “signal” and “noise”:

$$Y = G + N.$$

In this setting, we are now interested in showing that G is contained in the leading kernel PCA components, such that projecting G onto the leading kernel PCA components leads to only negligible error. In the following, we treat the signal and noise part of Y separately. This is possible because the projection π_d is a linear operation such that $\pi_d(Y) = \pi_d(G + N) = \pi_d(G) + \pi_d(N)$.

3. Also note that the capacity control typically employed in kernel methods amounts to some form of regularization, or “implicit denoising” (Smola et al., 1998). Therefore, we do not expect that the results using G are generally better than with the original labels. However, as we will see below, *unregularized* methods perform on par with kernel methods with capacity control using the estimated relevant information vector G .

3.2 The Relevant Information Vector

We first treat the relevant information vector G . The location of G with respect to the kernel PCA components is characterized by scalar products with the eigenvectors of the kernel matrix. We start by discussing this relationship in an asymptotic setting and then transfer the results back to the finite sample setting using convergence results for the spectral properties of the kernel matrix

Using the kernel function k , we define the integral operator

$$T_k f(s) = \int_{\mathcal{X}} k(s,t) f(t) P_{\mathcal{X}}(dt),$$

where $P_{\mathcal{X}}$ is the marginal distribution which generates the inputs X_i . It is well known that the linear operator

$$\tilde{T}_k f(s) = \frac{1}{n} \sum_{i=1}^n k(s, X_i) f(X_i)$$

represented by the kernel matrix approximates T_k as the number of points tend to infinity (see, for example, von Luxburg, 2004). While this follows easily for a fixed f and s , making the argument theoretically exact for operators (this means uniform over all functions) is not trivial.

As a consequence, the eigenvalues and eigenvectors of \tilde{T}_k , which are equal to those of the kernel matrix, converge to those of T_k (see Koltchinskii and Giné, 2000; Koltchinskii, 1998). In particular, scalar products of sample functions and eigenvectors of \mathbf{K} converge to scalar products with eigenfunctions of T_k . The asymptotic counterpart of the relevant information vector G is the function

$$g(x) = E(Y|X = x).$$

These correspondences are summarized in Figure 3. In summary, we can think of $z_i = u_i^\top G$ (properly scaled) as an approximation to $\zeta_i = \langle \psi_i, g \rangle$.

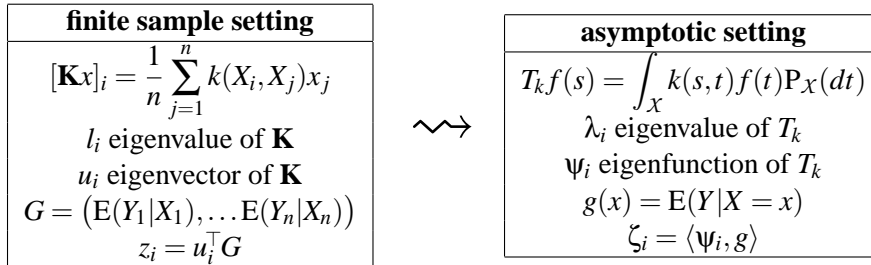


Figure 3: Transition from the finite sample size and asymptotic setting.

In the asymptotic setting, it is now fairly easy to specify conditions such that g is contained in the subspace spanned by a finite number of leading eigenfunctions ψ_i . Since it is unrealistic that g is exactly contained in a finite dimensional subspace, we relax that requirement and instead only require that ζ_i decays to zero at the same rate as the eigenvalues of T_k .

The decay rate of the eigenvalues depends on the interplay between the kernel and the distribution $P_{\mathcal{X}}$. However, expressing this connection in closed form is in general not possible. As a rule of thumb, the eigenvalues decay quickly when the kernel is smooth at the scale of the data. Since one usually uses smooth kernels to prevent from overfitting, the eigenvalues typically decay rather quickly. As we will see, most of the information about g is then contained in a few kernel PCA components.

A natural assumption is that the learning problem can be asymptotically represented by the given kernel function k . By this we mean that there exists some function $h \in \mathcal{L}^2(\mathcal{X}, \mathcal{P}_{\mathcal{X}})$ such that $g = T_k h$. Using the spectral decomposition of T_k , this implies

$$g = T_k h = \sum_{i=1}^{\infty} \lambda_i \langle \psi_i, h \rangle \psi_i. \tag{2}$$

Since the sequence of $\alpha_i = \langle \psi_i, h \rangle$ is square summable, it follows that

$$\zeta_i = \langle \psi_i, g \rangle = \lambda_i \alpha_i = O(\lambda_i).$$

Intuitively speaking, (2) translates to asymptotic representability of the learning problem: As $n \rightarrow \infty$, it becomes possible to represent the optimal labels using the kernel function k .

Furthermore, we assume that k is bounded. This technical requirement is mainly necessary to ensure that g is also bounded. The requirement holds for common radial basis function kernels like the Gaussian kernel, and also if the underlying space \mathcal{X} is compact and the kernel is continuous.

Note that the requirement that g lies in the range of T_k is essential. If this is not the case, we cannot expect that the scalar products decay at a given rate. Also note that it is in fact possible to break this condition. For example, if k is continuous, every non-continuous function does not lie in the range of T_k .

The question is now whether the same behavior can be expected for a finite data set. This question is not trivial, because eigenvector stability is known to be linked to the gap between the corresponding eigenvalues, which is fairly small for small eigenvalues (see, for example, Zwald and Blanchard, 2006).

The main theoretical result of this paper (Theorem 1 in the Appendix) provides a bound of the form

$$\frac{1}{n} |u_i^\top G| \leq l_i C + E$$

which expresses an essential equivalence between the finite sample setting and the asymptotic setting with two modifications: The decay rate $O(\lambda_i)$ of the scalar products $\langle \psi_i, g \rangle$ holds for the finite sample up to a (small) additive error E with λ_i replaced by its finite sample approximation l_i .

The technical details of this theorem and the proof are deferred to the appendix. Let us discuss how the absolute term occurs in the bound and why it can be expected to be small. An exact scaling bound (without additive term E) can only be derived (at least following the approach taken in this paper) for the case where the kernel function is degenerate, that is, T_k has only finitely many non-zero eigenvalues. The same finiteness restriction also holds for the expansion of g in terms of the eigenfunctions of T_k . The proof thus contains a truncation step of general kernels and general functions g , leading to a scaling bound on the scalar product and an additive term arising from the truncation. However, as the name suggests, the truncation error E can be made arbitrarily small by considering approximations with many non-zero eigenvalues. At the same time, considering such kernels with more terms in the expansion leads to a larger constant C in the actual scaling part. Thus, both terms have to be balanced by the order of truncation, which permits to control the additive term well practically.

Note that the problem considered here is significantly different from the problem studying the performance of kernel PCA itself (see, for example, Blanchard et al., 2007; Shawe-Taylor et al., 2005; Mika, 2002). There, only the projection error using the X s is studied. Here, we are specifically interested in the relationship between the Y s and the X s.

In view of our original concern, the bound shows that the relevant information vector G (as introduced in Section 2) is contained in a number of leading kernel PCA components up to a negligible error. The number of dimensions depends on the asymptotic coefficients α_i and the decay rate of the asymptotic eigenvalues of k . Since this rate is related to the smoothness of the kernel function, the dimension is small for smooth kernels whose leading eigenfunctions ψ_i permit good approximation of g .

3.3 The Noise

To study the relationship between the noise and the eigenvectors of the kernel matrix, no asymptotic arguments are necessary. The key insight is that the eigenvectors are independent of the noise in the labels, such that the noise vector N is typically evenly distributed over all coefficients $u_i^\top N$: Let \mathbf{U} be the matrix whose i th column is equal to u_i . The coefficients of N with respect to the eigenbasis of \mathbf{K} are then given by $\mathbf{U}^\top N$. Note that since \mathbf{U} is orthogonal, multiplication by its transpose amounts to a (random) rotation. In particular, this rotation is independent of the noise N as the u_i depend on the X s only. Now if the noise has a spherical distribution, for example, N is normally distributed with covariance matrix $\sigma_\varepsilon^2 \mathbf{I}$, it follows that $\mathbf{U}^\top N \sim \mathcal{N}(0, \sigma_\varepsilon^2 \mathbf{I})$. For heteroscedastic noise in a regression setting, or for classification, this simple analysis is not sufficient. In that case, the individual $u_i^\top N$ are no longer uncorrelated. However, because of the independence of the N_i , the variance of $u_i^\top N$ is upper bounded by

$$\text{Var}(u_i^\top N) = \sum_{j=1}^n u_{i,j}^2 \text{Var}(N_j) \leq \max_{1 \leq j \leq n} \text{Var}(N_j)$$

since $\sum_{j=1}^n u_{i,j}^2 = \|u_i\|^2 = 1$. Therefore, the variance of the $u_i^\top N$ is not concentrated in any single coefficient as the total variance does not increase by rotating the basis and the individual variances are bounded by the maximum individual variance before the rotation.

The practical relevance of these observations is that the relevant information and noise part have radically different properties with respect to the kernel PCA components, allowing us to practically estimate the number of relevant dimension for a given kernel and data set. In the next section, we will propose two different algorithms for this task.

4. Relevant Dimension Estimation and Related Estimates

We have seen that the number of leading kernel PCA components necessary to capture the relevant information about the labels of a finite size data set is bounded under the mild assumptions that the learning problem can be represented asymptotically and the kernel is smooth such that the eigenvalues of the kernel matrix decay quickly. The actual number of necessary dimensions depends on the interplay between kernel and learning data set, giving insights into the suitability of the kernel. For example, a kernel might fail to provide an efficient representation of the learning problem, leading to an embedding requiring many kernel PCA components to capture the information on Y . Or, even worse, a kernel might completely fail to model some part of the learning problem, such that a part of the information appears to be just noise. Therefore, in order to make practical use of the presented insights, we need to devise a method to estimate the number of relevant kernel PCA components for a given concrete data set and choice of kernel.

In this section we propose methods for estimating the actual dimensionality of a data set, and two related estimators. Based on the dimensionality estimate, one can denoise the labels by projecting

onto the respective subspace and obtain an estimate for the relevant information vector G . By comparing the denoised labels with the original labels, one can then estimate the overall noise level of the data source. Based on these estimates, we discuss how to use the dimensionality estimate for model-selection and to further analyze data sets which so far show inferior performance. Figure 4 summarizes the information flow for the different estimates.

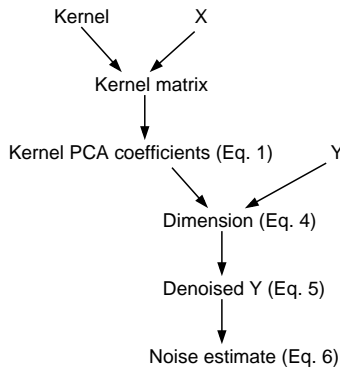


Figure 4: Information flow for the estimates.

4.1 Relevant Dimension Estimation (RDE)

The most basic estimate is the number of relevant kernel PCA components. We also call this number simply the *relevant dimension* or the *dimensionality* (also see the discussion in Section 6.3). Recall that we have decomposed the labels into $Y = G + N$, with $G_i = E(Y_i|X_i)$ (see Section 3.1). This decomposition carries over to the kernel PCA coefficients $z_i = u_i^\top Y = u_i^\top G + u_i^\top N$. We want to estimate \hat{d} such that $|u_i^\top G|$ is negligible for $i > \hat{d}$.

We propose two algorithms for solving this relevant dimension estimation (RDE) task which are based on different approaches to the problem but lead to comparable performance. The first algorithm fits a parametric model to the kernel PCA coefficients, while the second one is based on leave-one-out cross-validation.

4.1.1 RDE BY FITTING A TWO-COMPONENT MODEL (TCM)

The first algorithm works only on the coefficients $z_i = u_i^\top Y$. Recall that \mathbf{U} is the matrix whose columns are the eigenvectors of the kernel matrix u_i such that $z = \mathbf{U}^\top Y = \mathbf{U}^\top G + \mathbf{U}^\top N = \tilde{G} + \tilde{N}$. In Section 3, we have seen that both parts have significantly different structure. From Theorem 1, we know that $|\tilde{G}_i| \approx O(l_i)$, and that the \tilde{G}_i are close to zero for all but a leading number of coefficients. On the other hand, as discussed in Section 3.3, the transformed noise \tilde{N} is typically evenly distributed over all coefficients. Thus, the coefficients of the noise have the shape of an evenly distributed “noise floor” \tilde{N}_i from which the coefficients \tilde{G}_i of the relevant information arise (see Figure 1(b) for an example).

The idea is now to find a cut-off point such that the coefficients are divided into two parts z_1, \dots, z_d and z_{d+1}, \dots, z_n such that the first part contains the relevant information and the latter part consists of evenly distributed noise. We model the coefficients by two zero-mean Gaussians with

individual variances

$$z_i \sim \begin{cases} \mathcal{N}(0, \sigma_1^2) & 1 \leq i \leq d, \\ \mathcal{N}(0, \sigma_2^2) & d < i \leq n. \end{cases}$$

Of course, in order to be able to extract meaningful information, it should hold that $\sigma_1 \gg \sigma_2$. Alternatively, one could assume that $z_i \sim \mathcal{N}(0, \sigma_1^2 + \sigma_2^2)$, for $1 \leq i \leq d$, which nevertheless leads to the exact same choice of d .

For real data, both parts need not be actually Gaussian distributed. However, due to lack of additional a priori knowledge on the signal or the noise, the Gaussian distribution represents the optimal choice among all distributions with the same variance according to the maximum entropy principle (Jaynes, 1957).

The negative log-likelihood is proportional to

$$-\log \ell(d) \sim \frac{d}{n} \log \sigma_1^2 + \frac{n-d}{n} \log \sigma_2^2, \quad \text{with} \quad \sigma_1^2 = \frac{1}{d} \sum_{i=1}^d z_i^2, \quad \sigma_2^2 = \frac{1}{n-d} \sum_{i=d+1}^n z_i^2. \quad (3)$$

The estimated dimension is then given as the maximum likelihood fit

$$\hat{d} = \operatorname{argmin}_{1 \leq d \leq n'} (-\log \ell(d)) = \operatorname{argmin}_{1 \leq d \leq n'} \left(\frac{d}{n} \log \sigma_1^2 + \frac{n-d}{n} \log \sigma_2^2 \right). \quad (4)$$

Due to numerical instabilities of kernel PCA components corresponding to small eigenvalues, the choice of d should be restricted to $1 \leq d \leq n' < n$: The coefficients z_i are computed by taking scalar products with eigenvectors u_i . For small eigenvalues (small meaning of the order of the available numerical precision, for double precision floating point numbers, this is typically around 10^{-16}), individual eigenvectors cannot be computed accurately, although the space spanned by all these eigenvectors is accurate. Therefore, coefficients z_i for large i are not be reliable. To systematically stabilize the algorithm, one should therefore limit the range of possible effective dimensions. We have found the choice of $1 \leq d \leq n/2$ to work well as this choice ensures that at least half of the coefficients are interpreted as noise. For very small and very complex data sets, this choice might prove suboptimal and better thresholds based, for example, on the actual decay of eigenvalues might be advisable. However, on all data sets discussed in this paper, the above choice performed very well.

4.1.2 RDE BY LEAVE-ONE-OUT CROSS-VALIDATION (LOO-CV)

We propose a second algorithm which is based on cross validation, a more general concept than parametric noise modeling. This algorithm only depends on our theoretical results to the extent that it searches for subspaces spanned by leading kernel PCA components. We later compare the two methods to see whether our assumptions were justified.

As stated in Lemma 1, the projection of Y onto the space spanned by the d leading kernel PCA components is given by $\sum_{i=1}^d u_i u_i^\top Y$, where u_i are the eigenvectors of the kernel matrix. The matrix $\mathbf{S} = \sum_{i=1}^d u_i u_i^\top$ can be interpreted as a “hat matrix” in the context of regression.⁴ The idea is now to choose the dimension which minimizes the leave-one-out cross-validation error. This subspace then captures all of the relevant information about Y without overfitting.

4. Recall that for regression methods where the fitted function depends linearly on the labels, the matrix \mathbf{S} which computes $\hat{Y} = \mathbf{S}Y$ is called the “hat matrix” since it “puts the hat on Y .”

Computationally, note that one can write the squared error leave-one-out cross-validation in closed form, similar to kernel ridge regression (see Wahba, 1990):

$$\text{cv}_{\text{loo}}(d) = \frac{1}{n} \sum_{i=1}^n \left(\frac{[\mathbf{SY}]_i - Y_i}{1 - \mathbf{S}_{ii}} \right)^2.$$

It is possible to organize the computation in a way such that given the eigendecomposition of \mathbf{K} , each value $\text{cv}_{\text{loo}}(d)$ can be computed in $O(n)$ (instead of $O(n^2)$ if one naively implements the above formula): Note that \mathbf{S}_{ii} is equal to $\sum_{j=1}^d (u_j)_i^2$, therefore, one can compute \mathbf{S}_{ii} iteratively by

$$\begin{aligned} \mathbf{S}_{ii}^0 &\leftarrow 0 \\ \mathbf{S}_{ii}^{d+1} &\leftarrow \mathbf{S}_{ii}^d + (u_{d+1})_i^2. \end{aligned}$$

In the same way, since $\hat{Y} = \mathbf{SY} = \sum_{j=1}^d u_j u_j^\top Y$, we get that

$$\begin{aligned} \hat{Y}^0 &\leftarrow 0 \\ \hat{Y}^{d+1} &\leftarrow \hat{Y}^d + u_{d+1} u_{d+1}^\top Y. \end{aligned}$$

The squared error is in principle not the most appropriate loss function for classification problems. But as we will see below, it nevertheless works well also for classification problems.

4.2 Denoising the Labels and Estimating the Noise Level

One direct application of the dimensionality estimate is the projection of Y onto the first \hat{d} kernel PCA components. By Lemma 1, this projection is

$$\hat{G}' = \sum_{i=1}^{\hat{d}} u_i u_i^\top Y.$$

Then, an estimate of the noiseless labels is given by

$$\hat{G} = \begin{cases} \text{sign } \hat{G}' & \text{classification against } \pm 1 \text{ labels} \\ \hat{G}' & \text{regression} \end{cases}. \quad (5)$$

Note that this amounts to computing the in-sample fit using kernel principal component regression (kPCR).

The estimated dimension can also be used to estimate the noise level present in the data set by

$$\hat{\text{er}} = \frac{1}{n} \sum_{i=1}^n L(\hat{Y}_i, Y_i), \quad (6)$$

where L is the loss function.

The accuracy of both these estimates depends on a number of factors. Basically, the estimation error is small if the first \hat{d} kernel PCA components capture most of G and \hat{d} is small such that most of the noise is removed. Note that our assumption that the kernel suits the data set is crucial for both these requirements. If g does not lie in the span of the associated integral operator T_k , the coefficients decay only slowly and a huge number of dimensions are necessary to capture most information about G , leading to a huge amount of residual noise.

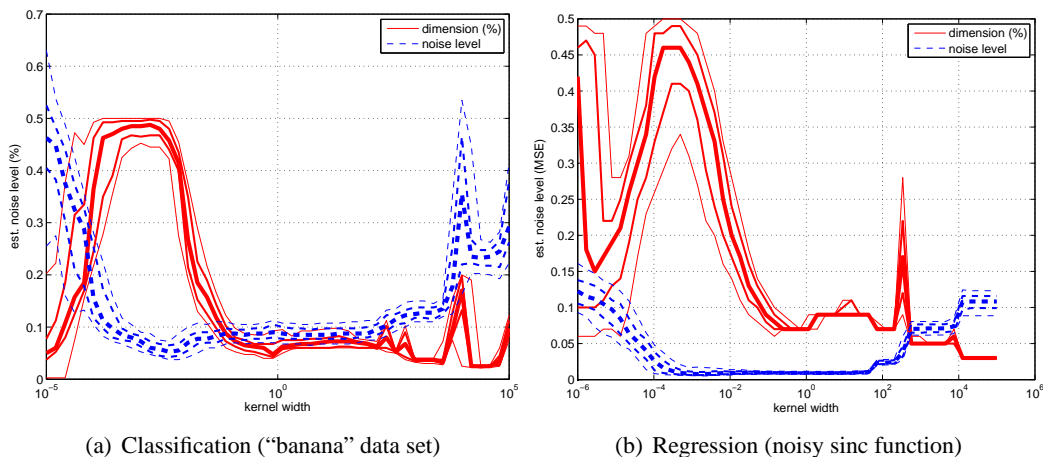


Figure 5: Dimensions and estimated noise levels for varying kernel widths are not suited for model selection as it is unclear how to combine both estimates and they become unstable for very small kernel widths. Shown are the 10%, 25%, 50%, 75%, and 90% percentiles over 100 resamples. **Legend:** “dimension (%)”—estimated dimensionality divided by number of samples. “noise level”—estimated noise level using the ℓ_1 -norm for classification, and the (unnormalized) ℓ_2 -norm for regression.

4.3 Applications to Model Selection

A highly relevant problem in the context of kernel methods is the selection of a kernel from a number of possible candidates which fits the problem best. This problem is usually solved by extensive cross-validation.

We would like to discuss possibilities to use the estimates introduced so far for model selection. Choosing the model based on either dimensionality or noise level alone is not sufficient, since one wants to optimize a combination of both. However, as the two terms live on quite different scales, it is unclear how to combine them effectively. Furthermore, as we will see below, both estimates alone become unstable for very small or very large kernel widths. The log-likelihood which achieves the optimum in (4) overcomes both problems and can be used for effective model selection.

Let us first discuss how the relation between the scale of the kernel and the data set can affect the dimensionality of the embedding in feature space. The standard example for a family of kernels with a scale parameter is the rbf-kernel (also known as Gaussian kernel, see Appendix A). Figure 5 shows the dimension and noise level estimates for a classification data set (the “banana” data set), and a regression data set (the “noisy sinc function” with 100 data points for training, and 1000 data points for testing) over a range of kernel widths. Generally speaking, if the scale of the kernel is too coarse for the problem, the problem tends to appear to be very low-dimensional with a large amount of noise. On the other hand, if the scale of the kernel is too fine the learning problem appears to be very complex with almost no noise.

Now, the log-likelihood $\ell(\hat{d})$ solves both problems. It combines the dimension and the noise level into a single meaningful number, and its value is stable across the whole scale range. In Figure 6, we have plotted the log-likelihood (scaled to fit into the plot) against the test error, both

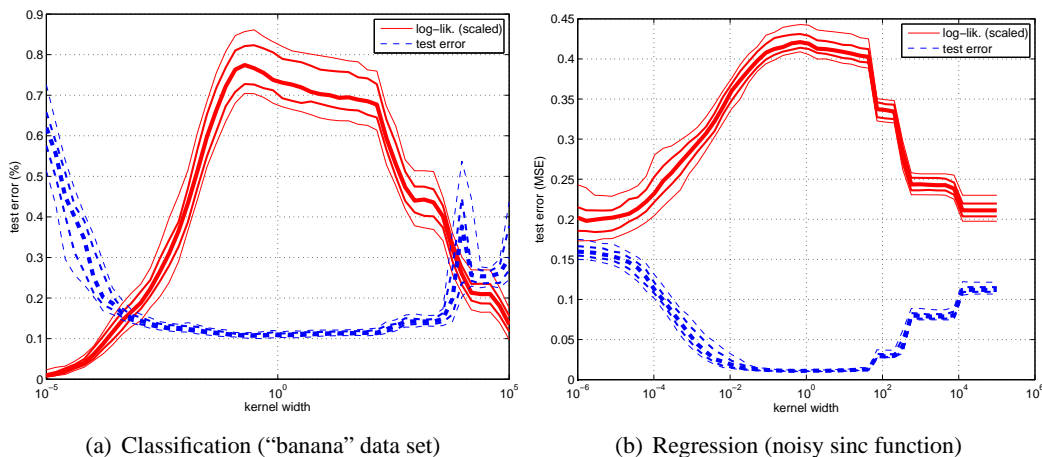


Figure 6: Comparison of test errors and the negative log-likelihood from Equation (3) shows that the negative log-likelihood is highly correlated with the test error and can thus be used for model selection. Shown are the 10%, 25%, 50%, 75%, and 90% percentiles over 100 resamples. **Legend:** “log-lik. (scaled)”—log-likelihood (scaled). “test error”—test error using the ℓ_1 -norm for classification, or the (unnormalized) ℓ_2 -norm for regression.

with respect to the classification and least squares error. We see that the estimated log-likelihoods can be estimated well over the whole range, and that the likelihoods are highly correlated with the actual test error. Thus, the log-likelihood is a reliable indicator for the test errors based on the best separation between signal and noise.

Another alternative, which is somewhat more straight-forward, but conceptually also less interesting, is to use the leave-one-out cross-validation error. This quantity also measures how well the kernel can separate the noise from the relevant information, and is directly linked to the test error on an independent data set. We validate both model selection approaches experimentally in Section 5.

4.4 Applications to Data Set Assessment

When working on a concrete data set in a kernel setting, one is faced with the problem of finding a suitable kernel. This problem is usually approached with a mix of hard-won experience and domain knowledge. The main tool for guiding the search are prediction performance measures, the classical one being prediction accuracy. Measurements like the ROC (receiver-operator-curve), or the AUC (area-under-the-curve) give more fine-grained measurements of prediction quality, in particular in areas where many false positives or false negatives are not acceptable.

If, after testing a number of sensible candidates, the achieved prediction quality is satisfying, this approach is perfectly adequate, but more often than not, prediction quality is not as good as desirable. In such a case, it is important to identify the cause for the inferior performance. In principle, three alternatives are possible:

1. The kernels which have been used so far are not suited for the problem.
2. The learning problem is very complex and requires more data.

data set	RDE method	dimension	noise-level
complex data set	TCM	50	16.07%
	LOO-CV	25	40.59%
noisy data set	TCM	9	40.71%
	LOO-CV	9	40.71%

Table 2: Estimated dimensions for the two data sets from Figure 7. Methods are “TCM” for RDE by fitting a two-component model, “LOO-CV” for RDE by leave-one-out cross-validation. “noise-level” is measured as normalized mean square error (see Appendix A).

3. Better performance cannot be achieved since the learning problem is intrinsically noisy.

Each of these alternatives requires different approaches. In the first case, a better kernel has to be devised, in the second case, more data has to be acquired, and in the last case, one can either stop searching for a better kernel, or try to improve the quality of the data or the features used.

Ultimately, these questions cannot be answered without knowledge of the true distribution of the data, but the important observation here is that performance measures do not provide enough information to distinguish these cases.

The estimates introduced so far can now be used to obtain evidence for distinguishing between the second and third case. On the one hand, the dimensionality of the problem is related to the complexity of the problem, while the noise level measures the inherent noise. Note that both these estimates depend on the chosen kernel.

Consider the following example: We study two data sets, a simple data set built from a noisy sinc function, and a complex data set based on a high-frequency sine function (see Figure 7). For the same number of data points $n = 100$, both data sets lead to comparable normalized test errors⁵ for the best model selected (A normalized test error of 43.7% on the complex data set and 44.4% on the noisy data set using kernel ridge regression with model selection by leave-one-out cross-validation. Widths were selected from 20 logarithmically spaced points from 10^{-6} to 10^2 , regularization constant was selected from 10 logarithmically spaced points from 10^{-6} to 10^3). However, the reason for the large error on the complex data set is clearly due to the small number of samples. If we increase the data set size to 1000 points, the normalized test error becomes 2.4%.

The question is now whether we can distinguish these two cases based on the kernel PCA coefficients. In fact, even on visual inspection, the kernel PCA coefficients display significant differences (see Figures 7(c) and 7(d)). We estimate the effective dimension and the resulting noise-level using the two methods we have proposed, the results are shown in Table 2. While both methods lead to different estimates, they both agree on the fact that the noisy data set has comparably low complexity and high noise, while the complex data set is quite high-dimensional, in particular if one takes into account that the data set contains only 100 data points. In fact, the RDE analysis on the larger complex data set with 1000 data points gives a dimension of 142, and a noise-level of 1.96%. Thus, the RDE measure correctly indicates that the large test error is due to the insufficient amount of data in the one case, and due to the large noise level in the other case.

This simple example demonstrates how the RDE measure can provide further information beyond the error rates. Below, we discuss this approach for several benchmark data sets.

5. See Appendix A for a definition of the normalized error.

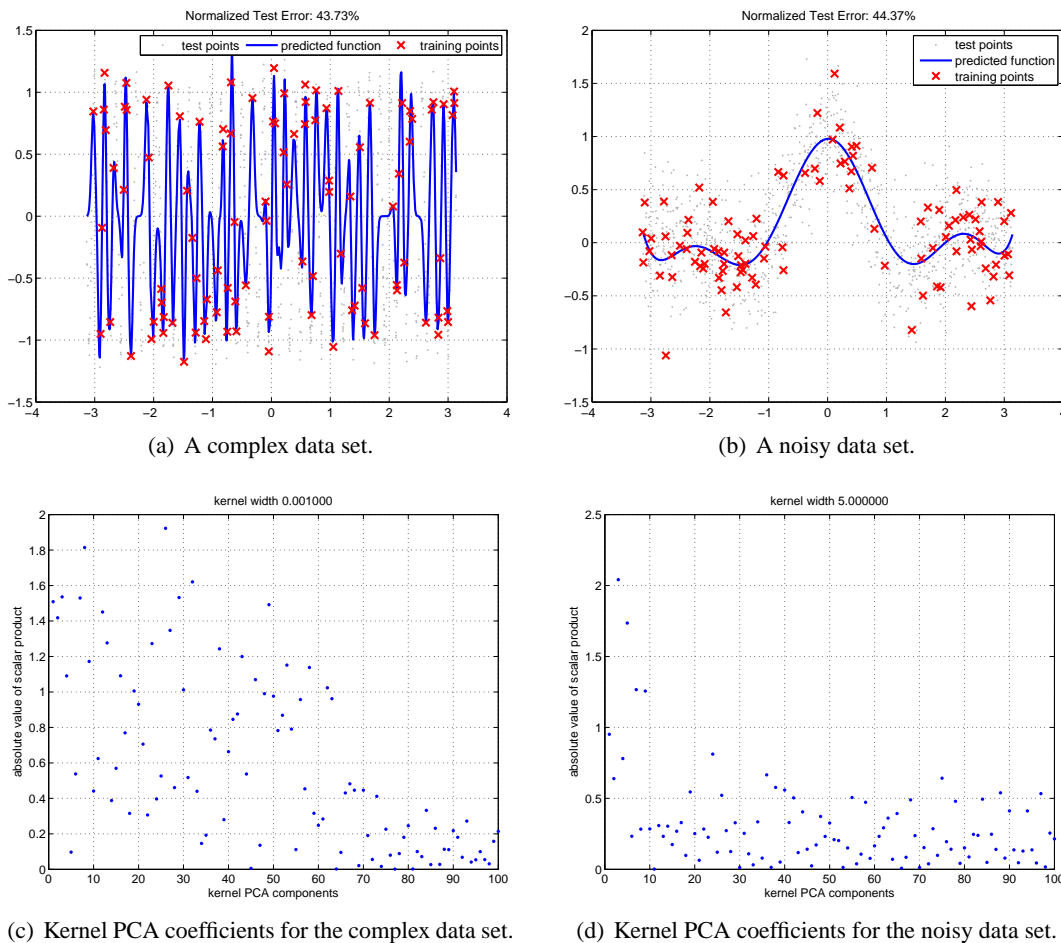


Figure 7: For both data sets, the X values were sampled uniformly between $-\pi$ and π . For the complex data set, $Y = \sin(35X) + \epsilon$ where ϵ has mean zero and variance 0.01. For the noisy data set, $Y = \text{sinc}(X) + \epsilon'$ where ϵ' has mean zero and variance 0.09. Errors are reported as normalized mean squared error (see Appendix A). Below, the kernel PCA coefficients (scalar products with eigenvectors of the kernel matrix) for the optimal kernel selected based on the RDE (TCM) estimates are plotted. Coefficients are sorted by decreasing corresponding eigenvalue.

5. Experiments

We test our methods on several benchmark data sets. As discussed in the introduction, in order to validate whether our dimension estimates are accurate, we compare the achieved test error rates on the reduced feature space to other state-of-the-art algorithms. If the estimate is accurate, the test errors should be on par with these algorithms. Furthermore, we apply our method to estimate the complexity and noise level of the various data sets.

5.1 Benchmark Data Sets

We performed experiments on the classification data sets from Rätsch et al. (2001). For each of the data sets, we analyze it using a family of rbf kernels (see Appendix A). The kernel width is selected automatically using the achieved log-likelihood as described above. The width of the rbf kernel is selected from 20 logarithmically spaced points between 10^{-2} and 10^4 for each data set.

Table 3 shows the resulting dimension estimates using both RDE methods, with the cross-validation based RDE method being slightly biased towards higher dimensions. We see that both methods perform on par, which shows that the strong structural prior assumption underlying RDE is justified.

To assess the accuracy of the dimensionality estimate, we compare an unregularized least-squares fit in the reduced feature space (RDE+kPCR) with kernel ridge regression (KRR) and support vector machines (SVM) on the original data set. The resulting test errors are also shown in Table 3. Note that the combination of RDE and kPCR is conceptually very similar to the kernel projection machine (Vert et al., 2005) which also produces comparable results. However, in that paper, no practical method for estimating the dimension (beyond cross-validation) has been proposed. From the resulting test errors, we see that a relatively simple method on the reduced features performs on par with the state-of-the-art competitors. We conclude that the identified reduced feature space really contains all of the relevant information. Also note that the estimated noise levels match the actually observed error rates quite well, although there is a slight tendency to under-estimate the true error.

As discussed in Section 4.4, while the test errors only suggest a linear ordering of the data sets by increasing difficulty, using the dimension and noise level estimates, a more fine-grained analysis is possible. We can roughly divide the data sets into four classes (see Table 4), depending on whether the dimensionality is small or large, and the noise level is low or high. Data sets with small noise level show good results, almost irrespective of the dimensionality. The data set *image* seems to be particularly noise free, given that one can achieve a small error in spite of the large dimensionality.

The data sets *breast-cancer*, *diabetes*, *flare-solar*, *german*, and *titanic*, which all have test errors of 20% or more, have only moderately large dimensionalities. This means that the complexity of the underlying optimal decision boundary is not overly large (at least when viewed through the lens of the rbf-kernel), but a large inherent noise level prevents better results. Since this holds for rbf-kernels over a wide range of kernel widths, these results can be taken as a strong indicator that the Bayes error is in fact large.

The *splice* data set seems to be a good candidate for improvement. The noise level is moderately high, while the dimensionality with respect to the rbf-kernel seems quite high. We would like to use our dimensionality and noise level estimate as a tool to examine different kernel choices. (See Section C for further details).

Closer inspection of the data set reveals that a plain rbf-kernel is a suboptimal choice. The task of the splice data set consists in predicting whether there is a *splice-site* in the middle of a string of DNA (such sites encode the beginning and endings of coding regions on the DNA). In the data set, the four amino-acids A, C, G, T are encoded as numbers 1, 2, 3, and 4. Therefore, an rbf-kernel incorrectly assumes that C and G are more similar than A and T. One alternative which is more suited to this data set consists in encoding A, C, G, and T as binary four-vectors. The resulting kernel matrix has much smaller dimension, and also a smaller error rate (see Table 5).

data set	TCM	LOO-CV	TCM-noise level	RDE+kPCR	KRR	SVM
banana	24	26	8.8 ± 1.5	11.3 ± 0.7	10.6 ± 0.5	11.5 ± 0.7
breast-cancer	2	2	25.6 ± 2.1	27.0 ± 4.6	26.5 ± 4.7	26.0 ± 4.7
diabetes	9	9	21.5 ± 1.3	23.6 ± 1.8	23.2 ± 1.7	23.5 ± 1.7
flare-solar	10	10	32.9 ± 1.2	33.3 ± 1.8	34.1 ± 1.8	32.4 ± 1.8
german	12	12	22.9 ± 1.1	24.1 ± 2.1	23.5 ± 2.2	23.6 ± 2.1
heart	4	5	15.8 ± 2.5	16.7 ± 3.8	16.6 ± 3.5	16.0 ± 3.3
image	272	368	1.7 ± 1.0	4.2 ± 0.9	2.8 ± 0.5	3.0 ± 0.6
ringnorm	36	37	1.9 ± 0.7	4.4 ± 1.2	4.7 ± 0.8	1.7 ± 0.1
splice	92	89	9.2 ± 1.3	13.8 ± 0.9	11.0 ± 0.6	10.9 ± 0.6
thyroid	17	18	2.0 ± 1.0	5.1 ± 2.1	4.3 ± 2.3	4.8 ± 2.2
titanic	4	6	20.8 ± 3.8	22.9 ± 1.6	22.5 ± 1.0	22.4 ± 1.0
twonorm	2	2	2.3 ± 0.7	2.4 ± 0.1	2.8 ± 0.2	3.0 ± 0.2
waveform	14	23	8.4 ± 1.5	10.8 ± 0.9	9.7 ± 0.4	9.9 ± 0.4

Table 3: Estimated dimensions and error rates for the benchmark data sets from Rätsch et al. (2001). **Legend:** “TCM”—medians of estimated dimensionalities over resamples using the RDE by TCM methods. “LOO-CV”—dimensionality estimated by leave-one-out cross-validation. “TCM-noise level”—estimated error rate using the estimated dimension. “RDE+kPCR”—test error using a least-squares hyperplane on the estimated subspace in feature space. “KRR”—kernel ridge regression with parameters determined by leave-one-out cross-validation. “SVM”—the original error rates from Rätsch et al. (2001). **Best** and *second best* results are highlighted.

	low noise	high noise
low dimensional	banana, thyroid, waveform	breast-cancer, diabetes flare-solar, german heart, titanic
high dimensional	image, ringnorm	splice

Table 4: The data sets by noise level and complexity.

Still, there is further room for improvement. Using a *weighted-degree kernel*, which has been specifically designed for this problem (Sonnenburg et al., 2005), we obtain even better results: While the dimension is again slightly larger (but still moderate compared to the number of 1000 training examples), the noise level is even smaller. The reason is that the weighted degree kernel weights longer consecutive matches on the DNA differently while the rbf kernel just compares individual matches. Again, learning hyperplanes on the subspace of the estimated dimension leads to classification results on the test sets which are close to those predicted by the error level estimate.

6. Discussion

We discuss some implications of our results to learning theory. In particular we show how the “standard picture” on kernels and feature spaces is extended by our results. With respect to practical

kernel	RDE	est. error rate	RDE+kPCR
rbf	87	9.4 ± 1.0	12.9 ± 0.9
rbf (binary)	11	7.1 ± 1.0	7.6 ± 0.7
wdk	29	4.5 ± 0.7	5.5 ± 0.7

Table 5: Different kernels for the splice data set (for fixed kernel width $w = 50$). **Legend:** “rbf”—plain rbf-kernel, “rbf (binary)”—rbf-kernel on A, C, G, T encoded in binary four-vectors, “wdk”—weighted degree kernel (Sonnenburg et al., 2005).

applications we explain the role of RDE as a diagnosis tool for kernels. We close by contrasting our notion of dimension with two closely related dimensions, the dimension of the minimal subspace necessary to capture the relevant information about a learning problem, and the dimension of the data sub-manifold.

6.1 Connections to Learning Theory

We start with some informal reasoning about our findings much like in the spirit of Vapnik (1995). Although our ideas are not developed to all formal details, they are intended to provide some interesting insights on extensions to the general statistical learning theory picture (see Figure 8). The standard picture (see, for example, Burges, 1998; Müller et al., 2001) can be summarized as follows: The learning problem is given in terms of a finite data set in $\mathcal{X} \times \mathcal{Y}$. The kernel k implicitly embeds \mathcal{X} in some (potentially) high-dimensional feature space \mathcal{F} via the feature map Φ . Now since the feature space can be high-dimensional, it is argued that one needs to employ some form of appropriate complexity control in order to be able to learn. A prominent example are large margin classifiers, leading to support vector machines. Other examples include penalization of the norm of the weight vectors, which relates to a penalization of the norm in the resulting reproducing kernel Hilbert space (RKHS).

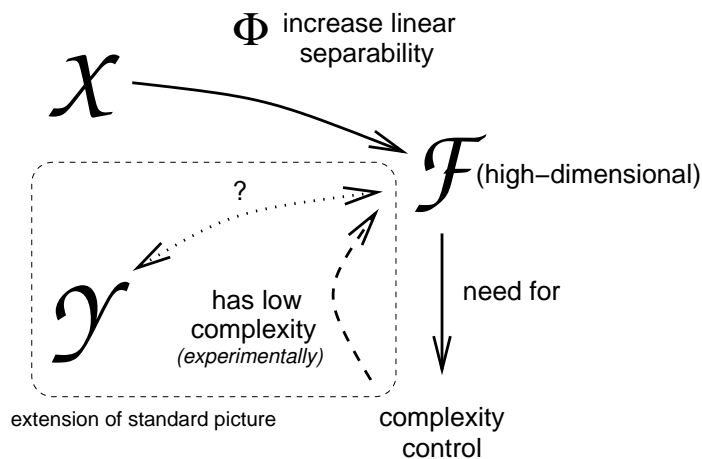


Figure 8: Learning in kernel feature spaces.

This picture is not entirely conclusive since it is not a priori clear that the feature map and the complexity control interact in a benign fashion. For example, it might be possible that the

feature map transforms the data such that a good representation can be learned, but the solution is incompatible with the kind of complexity one is penalizing. On the other hand, the large body of successful applications of kernel methods to real world problems is ample experimental verification of the fact that this seems to be the case and choosing a good kernel leads to an embedding which has low complexity, permitting, for example, large margin classifiers.

The question of the complexity of the image of X under the feature map actually has two parts. Part 1 concerns the complexity of the embedded object features $\Phi(X)$, while part 2 concerns the relation between the labels Y and the embedded object features $\Phi(X)$.

The first part has already been studied in several works. For example, Blanchard et al. (2007) and Braun (2006) have derived approximation bounds which show that the principal component values approximate the true principal values quickly (see also Mika, 2002; Shawe-Taylor et al., 2005). And since the asymptotic principal values decay rapidly, these results show that most of the variance of the X in feature space is contained in a finite dimensional subspace in feature space. Considering the function class generated by the feature map, Shawe-Taylor et al. (1998) first dealt with the complexity of kernel classes showing that the complexity can be bounded in the spirit of the structural risk minimization framework if a properly regularized class is picked depending on the data, for example by using large margin hyperplanes. Williamson et al. (2001) have further refined these results by using the concept of entropy numbers for compact operators that the complexity of the resulting hypothesis class is actually finite at any given positive scale. Evgeniou and Pontil (1999) show, using the concept of V_γ -dimension, which directly translates to a constraint on the RKHS-norm of the functions, that the resulting hypothesis classes have finite complexity. In summary, the embedding of X is known to have finite complexity (up to a small residual error).

The second part addresses the question if the embedding also relates favorably to the labels. In this work we have studied this question and answered it positively. One can prove that under mild assumptions on the general fit between the kernel and the learning problem, the information about the labels is always contained in the (typically small) subspace also containing most of the variance about the object features. While this borders on the trivial for the asymptotic setting, we could show that the same also holds true for a concrete finite data set, even at small sample sizes.

Our findings clarify the role of complexity control in feature space. The complexity control is not sufficient for effective learning in the feature space, but necessary. In conjunction with a sensible embedding provided by a suitable choice of the kernel function, it ensures that learning focuses on the relevant information and prevents overfitting. Interestingly, RKHS type penalty terms automatically ensure that the learned function focuses on directions in which the data has large variance, automatically leading to a concentration on the leading kernel PCA components.

6.2 RDE as a Diagnosis Tool

As discussed in Section 4.4, performance measures like the test error are very useful to compare different kernels, but fail to provide evidence if the performance is not as good as desired on whether the right kernel has not been found yet or the problem is intrinsically noisy.

Now, the RDE based estimates proposed in this paper offer a possible new approach to solve this problem. The relevant dimensionality estimate and the noise level estimate allow us to directly address the complexity vs. randomness issue, at least for a given kernel. Of course, our approach only provides a partial answer. However, using a generic kernel like an rbf-kernel for different widths results in an analysis of the data set on a whole range of scale resolutions. If the data set

appears to be low-dimensional and noisy at every scale, there is a strong indication that the noise level is actually quite high.

In the data sets discussed in Section 5, we have considered kernel widths in the range 10^{-2} to 10^4 . The data sets *breast-cancer*, *diabetes*, *flare-solar*, *german*, *heart*, and *titanic*, which all have prediction errors larger than 15%, turn out to be fairly low-dimensional over the whole range.

On the other hand, the splice data set seemed to be quite complex, but not very noisy. Using domain knowledge, we improved the encoding, and finally chose a different kernel, which further reduced the complexity and noise (see Section C for further details).

In summary, using the RDE based estimates as a diagnosis tool, it is possible to obtain more detailed insights into how well a kernel is adapted to the characteristic properties of a data set and its underlying distribution than by using integrative performance measures like test errors only.

6.3 The “True” Dimensionality of the Data

We estimate the number of leading kernel PCA components necessary to capture the relevant information contained in the learning problem. This “relevant dimensionality estimate” captures only a very special kind of dimensionality notion, and we would like to compare it with two other aspects of dimensionality.

In our dimensionality estimate, the basis was fixed and given by leading kernel PCA components. One might wonder how many dimensions are necessary to capture the relevant information about the learning problem if one were also allowed to choose the basis. The answer is easy: In order to capture G , it suffices to consider the one-dimensional space spanned by G itself, which means that the minimal dimensionality of the learning problem is 1. However, note that G is not known, and estimating G amounts to solving the learning problem itself. In other words, the choice of a kernel can be interpreted as implicitly specifying an appropriate basis in feature space which is able to capture G using as few basis vector as possible, *and* also using a subspace which contains as much of the variance of the data as possible.

For most data sets, the different input variables are highly dependent, such that the data does not occupy all of the space but only a sub-manifold in the space. The dimension of this sub-manifold is a further notion of dimensionality of a data set. However, note that we consider the dimensionality of the data with respect to the information in the labels, while the sub-manifold view usually concentrates on the inputs only. Also, we are considering linear subspaces (in an RKHS), which typically require more dimensions to capture the data than a non-linear manifold would. On the other hand, since we are only looking at the subspace which is relevant for predicting the labels, the estimated dimension may also be smaller than the dimension of the data manifold in feature space.

7. Conclusion

Both in theory and on practical data sets, we have demonstrated that the relevant information in a supervised learning scenario is contained in the leading projected kernel PCA components if the kernel matches the learning problem and is sufficiently smooth. This behavior complements the common statistical learning theoretical view on kernel based learning adding insight on the intricate interplay of data and kernel: An appropriately selected kernel (a) leads to an efficient model which generalizes well, since only a comparatively low dimensional representation has to be learned for a

fixed given data size. An appropriately selected kernel (b) permits a dimension reduction step that discards some irrelevant projected kernel PCA directions and thus yields a regularized model.

We propose two algorithms for the relevant dimensionality estimate (RDE) task. These can also be used to automatically select a suitable kernel model for the data and to extract as additional side information an estimate of the effective dimension and estimated expected error for the learning problem. Compared to common cross-validation techniques one could argue that all we have achieved is to find a similar model as usual at a comparable computing time. However, we would like to emphasize that the side information extracted by our procedure contributes to a better understanding of the learning problem at hand: Is the classification result limited due to intrinsic high dimensional structure or are we facing noise and nuisance dimensions? Simulations show the usefulness of our RDE algorithms.

An interesting future direction lies in combining these results with generalization bounds which are also based on the notion of an effective dimension, this time, however, with respect to some regularized hypothesis class (see, for example, Zhang, 2005). Linking the effective dimension of a data set with the “dimension” of a learning algorithm, one could obtain data dependent bounds in a natural way with the potential to be tighter than bounds which are based on the abstract capacity of a hypothesis class.

Acknowledgments

Parts of this work have been performed while MLB was with the Intelligent Data Analysis Group at the Fraunhofer Institute FIRST. The authors would like to thank Volker Roth, Tilman Lange, Gilles Blanchard, Stefan Harmeling, Motoaki Kawanabe, Claudia Sannelli, Jan Müller, and Nicole Krämer for fruitful discussions. The authors would also like to thank the anonymous referees whose comments have helped to improve the paper further, and in particular Peter Bartlett for his valuable comments. This work was supported in part by the BMBF FaSor project, 16SV2234, and by the FP7-ICT Programme of the European Community, under the PASCAL2 Network of Excellence, ICT-216886.

Appendix A. Data Sets and Kernel Functions

In this section, we introduce some data sets and define the Gaussian kernel, since there exists some variability with respect to its parameterization.

A.1 Gaussian kernel

The Gaussian kernel, or rbf-kernel, used in this paper are parameterized as follows: The Gaussian with width w is

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2w}\right).$$

A.2 Classification Data Sets

For classification, we use the data sets from Rätsch et al. (2001). This benchmark data set consists of 13 classification data sets, which are partly synthetic, and partly derived from real-world data. The data sets are pre-arranged into different resamples of training and test data sets. The number

of resamples is 100 with the exception of the “image” and “splice” data sets which have only 20 resamples (because these data sets are fairly large). For visualization purposes, we often take the first resample of the “banana” data set, which is a two-dimensional classification problem (see Figure 1(a)).

A.3 Regression Data Sets

The “noisy sinc function” data set is defined as follows:

$$\begin{aligned} X_i &\sim \text{uniformly from } [-\pi, \pi], \\ Y_i &= \text{sinc}(X_i) + \varepsilon_i, \\ \varepsilon_i &\sim \mathcal{N}(0, \sigma_\varepsilon^2). \end{aligned}$$

There are different alternatives for defining the sinc function, we choose $\text{sinc}(x) = \sin(\pi x)/\pi x$, $\text{sinc}(0) = 1$.

For regression, we sometimes measure the error using the “normalized mean squared error.” If the original labels are given by Y_i , $1 \leq i \leq n$, and the predicted ones are \hat{Y}_i , then this error is defined as

$$\text{nmse} = \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \frac{1}{n} \sum_{j=1}^n Y_j)^2}.$$

Appendix B. Proof of the Main Theorem

In this section, the main theorem of the paper is stated and proven. We start with some definitions, then introduce and discuss the assumptions of the main result. Next we define a few quantities on which the bound depends. The bound itself is split into two theorems. First the general bound is derived and then the asymptotic rates of these quantities are studied.

B.1 Preliminaries

Using the probability measure P_X which generates the X s, we can define a scalar product via $\langle f, g \rangle = \int_{\mathcal{X}} f(x)g(x)P_X(dx)$ which induces the Hilbert space $\mathcal{L}^2(\mathcal{X}, P_X)$. Unless indicated otherwise, $\|f\|$ will denote the norm with respect to this scalar product. Let $k(x, y) = \sum_{\ell=1}^{\infty} \lambda_\ell \psi_\ell(x)\psi_\ell(y)$ be a kernel function (such that $\lambda_\ell \geq 0$). The ψ_ℓ form an orthogonal family of functions on the Hilbert space $\mathcal{L}^2(\mathcal{X}, P_X)$. Given an n -sample X_1, \dots, X_n from P_X , the sample vector of a function g is the vector $g(\mathbf{X}) = (g(X_1), \dots, g(X_n))$. The kernel matrix given a kernel function k and an n -sample X_1, \dots, X_n is the $n \times n$ matrix \mathbf{K} with entries $k(X_i, X_j)/n$.

Let $g(x) = \sum_{\ell=1}^{\infty} \alpha_\ell \lambda_\ell \psi_\ell(x)$ with $(\alpha_\ell) \in \ell^2$, the set of all square-summable sequences. The expansion of g in terms of $\lambda_\ell \psi_\ell$ amounts to assuming that g lies in the range of the integral operator T_k defined by $T_k f = \int_{\mathcal{X}} k(\cdot, x)f(x)P_X(dx)$. Then, $g = T_k h$ with $h = \sum_{\ell=1}^{\infty} \alpha_\ell \psi_\ell$.

The act of truncating an object with an infinite expansion to its first r coefficients is so ubiquitous in the following that we introduce a generic notation. If k is a kernel function, \tilde{k} is the kernel function whose expansion has been reduced to the first r terms. Likewise, $\tilde{\mathbf{K}}$ is the kernel matrix induced by \tilde{k} . For a sequence $(\alpha_\ell) \in \ell^2$, $\tilde{\alpha}$ is the tuple consisting of the first r elements of the sequence. The sample vector matrices $\tilde{\Psi}$ is formed by the sample vector of the first r eigenvectors, that is, $\tilde{\Psi}_{ij} = \psi_j(X_i)/\sqrt{n}$, and $\tilde{\Lambda}$ is the diagonal matrix formed from the first r eigenvalues, such that $\tilde{\mathbf{K}} = \tilde{\Psi}\tilde{\Lambda}\tilde{\Psi}^\top$. Finally, \tilde{g} is obtained from g by truncating the expansion to the first r eigenfunctions.

The eigen-decompositions of the kernel matrix and the truncated kernel matrix (kernel matrix for the truncated kernel function) are

$$\mathbf{K} = \mathbf{U}\mathbf{L}\mathbf{U}^\top, \quad \tilde{\mathbf{K}} = \tilde{\mathbf{U}}\tilde{\mathbf{L}}\tilde{\mathbf{U}}^\top,$$

where $\mathbf{U}, \tilde{\mathbf{U}}$ are orthogonal matrices with columns u_i, \tilde{u}_j , and $\mathbf{L}, \tilde{\mathbf{L}}$ are diagonal matrices with entries l_i, \tilde{l}_j , such that the eigenpairs of \mathbf{K} are (l_i, u_i) , and those of $\tilde{\mathbf{K}}$ are $(\tilde{l}_j, \tilde{u}_j)$. We stick to the general convention that eigenvalues are always sorted in decreasing order.

Tail-sums of eigenvalues are denoted by

$$\Lambda_{>r} = \sum_{i=r+1}^{\infty} \lambda_i, \quad \Lambda_{\geq r} = \sum_{i=r}^{\infty} \lambda_i.$$

We will refer to the following result relating decay rates of the eigenvalues to the tail-sums. For proofs, see, for example, Braun (2006). It holds that if $\lambda_r = r^{-d}$ with $d \geq 1$, then $\Lambda_{>r} = O(r^{1-d})$. If $\lambda_r = \exp(-cr)$ with $c > 0$, then $\Lambda_{>r} = O(\exp(-cr))$. The same rates hold for $\Lambda_{\geq r}$.

Furthermore, we will often make use of the fact that $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ if $a, b \geq 0$.

B.2 Assumptions

The overall goal is to derive a meaningful upper bound on $\frac{1}{\sqrt{n}}|u_i^\top g(\mathbf{X})|$. In particular, the bound should scale with the corresponding eigenvalue l_i . We proceed as follows: First, we derive the actual bound which depends on a number of quantities. In the next step, we estimate the worst case asymptotic rates of these quantities. The actual bound depends on assumptions which are discussed in the following.

(A1) We assume that the kernel is uniformly bounded, that is,

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

(A2) We assume that $n \geq r$ large enough such that $\tilde{\Psi}^\top \tilde{\Psi}$ is invertible.

(A3) We assume that $\lambda_i = O(i^{-5/2-\varepsilon})$ for some $\varepsilon > 0$.

Assumption (A1) is true for radial basis functions like the Gaussian kernel, but also if the underlying space \mathcal{X} is compact and the kernel is continuous. From (A1), it follows easily that g is bounded as well since

$$|g(x)| \leq K\|h\|.$$

Furthermore, since the ψ_i are orthogonal, it follows that $\|h - \tilde{h}\| \leq \|h\|$, and therefore

$$|g(x) - \tilde{g}(x)| \leq K\|h\|$$

since $g - \tilde{g} = T_k(h - \tilde{h})$, and therefore $|g(x) - \tilde{g}(x)| \leq K\|h - \tilde{h}\| \leq K\|h\|$. These inequalities play an important role for bounding the truncation error $g - \tilde{g}$ in a finite sample setting.

Since the sample vectors $\psi_\ell(\mathbf{X})$ are asymptotically pairwise orthogonal, $\tilde{\Psi}^\top \tilde{\Psi}$ converges to \mathbf{I} , and for large enough n , assumption (A2) is met. See also Lemma 2 below.

Assumption (A3) ensures that the term $r(\sum_{i=1}^r |\alpha_i|)\Lambda_{\geq r}$ occurring in the bound vanishes as $r \rightarrow \infty$. Note that since the sequence of α_i is square-summable,

$$\sum_{i=1}^r |\alpha_i| \leq \sqrt{r \sum_{i=1}^r \alpha_i^2} \leq \sqrt{r} \|\alpha\|_{\ell^2} = O(\sqrt{r}).$$

Therefore, $r \sum_{i=1}^r |\alpha_i| = O(r^{3/2})$. Also, $\Lambda_{\geq r} = O(r^{-3/2-\varepsilon})$, such that $r(\sum_{i=1}^r |\alpha_i|)\Lambda_{\geq r} = O(r^{-\varepsilon})$. Note that (A3) is quite modest and eigenvalues often decay much faster, even at exponential rates.

B.3 The Main Result

The following five quantities occur in the bound:

- $c_i = |\{1 \leq j \leq r \mid l_i/2 \leq \tilde{l}_j \leq 2l_i\}|$ is the number of eigenvalues of the truncated kernel matrix which are close to the eigenvalues of the normal kernel matrix. This is a measure for the approximate degeneracy of eigenvalues.
- $\tilde{a} = \|\tilde{\alpha}\|_1$, is a measure for the size of the first r coefficients which define g .
- $\tilde{\mathbf{E}} = \mathbf{K} - \tilde{\mathbf{K}}$ is the truncation error for the kernel matrix.
- $\tilde{T} = \|g - \tilde{g}\| = \sqrt{\sum_{j=r+1}^{\infty} \alpha_j^2 \lambda_j^2}$ is the asymptotic truncation error for the function g .
- $F = \|g\|_{\infty} < \infty$, an upper bound on g .

We study these quantities in more detail after proving the actual bound, which follows next.

Theorem 1 *With the definitions introduced so far, it holds that with probability larger than $1 - \delta$, for all $1 \leq i \leq n$,*

$$\frac{1}{\sqrt{n}} |u_i^\top g(\mathbf{X})| < \min_{1 \leq r \leq n} [l_i c_i D(r, n) + E(r, n) + T(r, n)]$$

where the three terms are given by

$$D(r, n) = 2\tilde{a} \|\tilde{\Psi}^+\|, \quad E(r, n) = 2r\tilde{a} \|\tilde{\Psi}^+\| \|\tilde{\mathbf{E}}\|, \quad T(r, n) = \tilde{T} + \sqrt{F\tilde{T}} \sqrt[4]{\frac{1}{n\delta}}.$$

Proof First, we replace $g = \tilde{g} + (g - \tilde{g})$ and obtain

$$\frac{1}{\sqrt{n}} |u_i^\top g(\mathbf{X})| \leq \frac{1}{\sqrt{n}} |u_i^\top \tilde{g}(\mathbf{X})| + \frac{1}{\sqrt{n}} \|g(\mathbf{X}) - \tilde{g}(\mathbf{X})\| =: (\text{I}),$$

using the Cauchy-Schwarz-inequality and the fact that $\|u_i\| = 1$ for the second term.

Next, we re-express $\tilde{g}(\mathbf{X}) = \sum_{\ell=1}^r \lambda_\ell \alpha_\ell \psi_\ell(\mathbf{X})$ as follows. By definition, $\tilde{g}(\mathbf{X})$ lies in the image of $\tilde{\mathbf{K}}$, therefore, $\tilde{g}(\mathbf{X}) = \sum_{j=1}^r \tilde{u}_j \tilde{u}_j^\top \tilde{g}(\mathbf{X})$. Using both these equations, we obtain

$$\frac{1}{\sqrt{n}} |u_i^\top \tilde{g}(\mathbf{X})| \leq \sum_{\ell=1}^r |\alpha_\ell| \sum_{j=1}^r (u_i^\top \tilde{u}_j) \left[\frac{1}{\sqrt{n}} \lambda_\ell \psi_\ell(\mathbf{X})^\top \tilde{u}_j \right] =: (\text{II})$$

The term $u_i^\top \tilde{u}_j$ measures the angle between the eigenvectors of \mathbf{K} and $\tilde{\mathbf{K}}$. Note that \mathbf{K} can be considered an additive perturbation of $\tilde{\mathbf{K}}$ by $\tilde{\mathbf{E}} = \mathbf{K} - \tilde{\mathbf{K}}$. Such perturbations are studied by the so-called *sin-theta-theorems*. Specializing Theorem 6.2 of Davis and Kahan (1970) (see Section D) to two single eigenvectors, we obtain that

$$|u_i^\top \tilde{u}_j| \leq \min\left(\frac{\|\tilde{\mathbf{E}}\|}{|l_i - \tilde{l}_j|}, 1\right).$$

The term $\lambda_\ell \psi_\ell(\mathbf{X})^\top \tilde{u}_j / \sqrt{n}$ is bounded by $\tilde{l}_j \|\tilde{\Psi}^+\|$ (where Ψ^+ denotes the pseudo-inverse of Ψ), since

$$\tilde{l}_j \tilde{u}_j = \tilde{\mathbf{K}} \tilde{u}_j = \tilde{\Psi} \tilde{\Lambda} \tilde{\Psi}^\top \tilde{u}_j \quad \Rightarrow \quad \tilde{l}_j \tilde{\Psi}^+ \tilde{u}_j = \tilde{\Lambda} \tilde{\Psi}^\top \tilde{u}_j.$$

Taking norms, we obtain $\|\tilde{\Lambda} \tilde{\Psi}^\top \tilde{u}_j\| \leq \tilde{l}_j \|\tilde{\Psi}^+\|$, from which the claimed inequality follows for each individual coordinate of the vector on the left-hand side.

Combining the bounds for the two terms $u_i^\top \tilde{u}_j$ and $\lambda_\ell \psi_\ell(\mathbf{X})^\top \tilde{u}_j / \sqrt{n}$, we obtain

$$(u_i^\top \tilde{u}_j) \left[\frac{1}{\sqrt{n}} \lambda_\ell \psi_\ell(\mathbf{X})^\top \tilde{u}_j \right] \leq \|\tilde{\Psi}^+\| \min\left(\frac{\|\tilde{\mathbf{E}}\|}{|l_i - \tilde{l}_j|}, 1\right) \tilde{l}_j =: \|\tilde{\Psi}^+\| c_{ij}.$$

For $j \notin J(l_i) = \{1 \leq j \leq r \mid \frac{1}{2}l_i \leq \tilde{l}_j \leq 2l_i\}$, it holds that $\|\tilde{\mathbf{E}}\| \tilde{l}_j / |l_i - \tilde{l}_j| \leq 2\|\tilde{\mathbf{E}}\|$, therefore,

$$\sum_{j=1}^r c_{ij} = \sum_{j \in J(l_i)} c_{ij} + \sum_{j \notin J(l_i)} c_{ij} \leq 2|J(l_i)|l_i + 2r\|\tilde{\mathbf{E}}\|.$$

We have just shown that

$$(\text{II}) \leq \|\tilde{\Psi}^+\| \sum_{\ell=1}^r |\alpha_\ell| (2|J(l_i)|l_i + 2r\|\tilde{\mathbf{E}}\|). \quad (7)$$

Now concerning the other term in (I), note that by the strong law of large numbers,

$$\frac{1}{n} \|g(\mathbf{X}) - \tilde{g}(\mathbf{X})\|_{\mathbb{R}^n}^2 \rightarrow \|g - \tilde{g}\|_{\mathcal{L}^2(X, P_X)}^2 = \sum_{j=r+1}^{\infty} \alpha_j^2 \lambda_j^2 =: \tilde{T}^2.$$

Since g is bounded, $\|g\|_\infty = F < \infty$, we can bound the variance of $g - \tilde{g}$:

$$\text{Var}_{P_X}((g - \tilde{g})^2) \leq \|g - \tilde{g}\|_\infty^2 \|g - \tilde{g}\|^2 = F^2 \tilde{T}^2.$$

We can thus bound the probability of a large deviation using the Chebychev-inequality. Taking the square roots, we obtain that with probability larger than $1 - \delta$,

$$\frac{1}{\sqrt{n}} \|g(\mathbf{X}) - \tilde{g}(\mathbf{X})\| \leq \tilde{T} + \sqrt{F\tilde{T}}(n\delta)^{-\frac{1}{4}}. \quad (8)$$

Combining bound (7) and (8), we obtain that

$$\frac{1}{\sqrt{n}} |u_i^\top g(\mathbf{X})| \leq 2l_i |J(l_i)| \|\tilde{\alpha}\|_1 \|\tilde{\Psi}^+\| + 2r \|\tilde{\mathbf{E}}\| \|\tilde{\alpha}\|_1 \|\tilde{\Psi}^+\| + \tilde{T} + \sqrt{F\tilde{T}}(n\delta)^{-\frac{1}{4}}.$$

This proves the upper bound on the coefficients. ■

B.4 Worst Case Asymptotic Rates of the Error Matrices

The bound depends on a number of error terms, whose worst case asymptotic rates and their dependency on r are studied next.

The norm of the pseudo-inverse of $\tilde{\Psi}$ can be related to the matrix $\tilde{\mathbf{C}} = \tilde{\Psi}^\top \tilde{\Psi} - \mathbf{I}$, which measures the deviation from orthonormality of the sample vectors of the first r eigenfunctions of T_k . Since the eigenfunctions are asymptotically orthonormal, it is guaranteed that $\|\tilde{\mathbf{C}}\| \rightarrow 0$ as $n \rightarrow \infty$.

Lemma 2 *Let $\tilde{\mathbf{C}} = \tilde{\Psi}^\top \tilde{\Psi} - \mathbf{I}$. If $\|\tilde{\mathbf{C}}\| < 1$, then*

$$\|\tilde{\Psi}^+\| \leq (1 - \|\tilde{\mathbf{C}}\|)^{-1/2} = 1 + O(\sqrt{\|\tilde{\mathbf{C}}\|}).$$

Proof Recall that $\|\tilde{\Psi}^+\| = 1/\sigma_r(\tilde{\Psi})$, where $\sigma_r(\tilde{\Psi})$ is the r th singular value of $\tilde{\Psi}$ in descending order. The singular values are the square roots of the eigenvalues of $\tilde{\Psi}^\top \tilde{\Psi}$, and

$$1 - \lambda_r(\tilde{\Psi}^\top \tilde{\Psi}) \leq \max_{1 \leq i \leq r} |\lambda_i(\tilde{\Psi}^\top \tilde{\Psi}) - 1| \leq \|\tilde{\Psi}^\top \tilde{\Psi} - \mathbf{I}\|,$$

and therefore $\sigma_r(\tilde{\Psi}) = (\lambda_r(\tilde{\Psi}^\top \tilde{\Psi}))^{1/2} \geq (1 - \|\tilde{\Psi}^\top \tilde{\Psi} - \mathbf{I}\|)^{1/2}$, which proves the inequality.

For the asymptotic rate, observe that

$$\|\tilde{\Psi}^+\| \leq \sqrt{\frac{1}{1 - \|\tilde{\mathbf{C}}\|}} = \sqrt{\frac{\|\tilde{\mathbf{C}}\|^{-1}}{\|\tilde{\mathbf{C}}\|^{-1} - 1}} = \sqrt{1 + \frac{1}{\|\tilde{\mathbf{C}}\|^{-1} - 1}} \leq 1 + \sqrt{\frac{1}{\|\tilde{\mathbf{C}}\|^{-1} - 1}}.$$

Now, $1/(x - 1) = O(1/x)$ for $x \rightarrow \infty$, which proves the asymptotic rate. ■

The two error matrices $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{E}}$ were discussed in depth by Braun (2006). However, note that these asymptotic rates are worst case rates over certain families of kernel functions. This means that the results on the asymptotic rates do not describe typical behavior but rather worst case behavior, and their main purpose of these rates is to ensure that the error terms cannot diverge rather than giving realistic estimates.

The following result is Theorem 4 from Braun (2006).

Lemma 3 *For $1 \leq r \leq n$, with probability larger than $1 - \delta$,*

$$\|\tilde{\mathbf{C}}\| < r \sqrt{\frac{r(r+1)K}{\lambda_r n \delta}}, \quad \|\tilde{\mathbf{E}}\| < \lambda_r + \Lambda_{>r} + \sqrt{\frac{2K\Lambda_{>r}}{n\delta}}.$$

From Lemma 3, it follows that

$$\begin{aligned} \|\tilde{\Psi}^+\| &= 1 + O(r\lambda_r^{-1/4} n^{-1/4}), \\ \|\tilde{\mathbf{E}}\| &= \Lambda_{\geq r} + O(\sqrt{\Lambda_{>r} n^{-1/2}}). \end{aligned}$$

If we plug these rates into the bound from Theorem 1 and suppress all parts which converge to zero, the bound becomes

$$\frac{1}{\sqrt{n}} |u_i^\top g(\mathbf{X})| \leq 2c_i \tilde{a} l_i + 2r \tilde{a} \Lambda_{\geq r} + \tilde{T} + \text{terms which vanish as } n \rightarrow \infty.$$

We see that the general structure of the bound consists a part which scales with the eigenvalue under consideration and an additive part which is independent of i . The factor of the scaling part increases with r since $\tilde{a} = O(\sqrt{r})$ in the worst case. At the same time, the truncation error \tilde{T} arising from the truncation of g becomes smaller as r is increased, and by assumption (A3), it is ensured that the second term actually converges to zero as $r \rightarrow \infty$. The two parts therefore form a trade-off and by choosing r , one can balance these two terms.

Now, in particular the convergence of $\|\tilde{\Psi}^+\| \rightarrow 1$ can be quite slow in the worst case, if the eigenvalues of the kernel matrix decay quickly (see the paper by Braun, 2006, for a more thorough discussion including an artificial example of a kernel function which achieves the described rate). However, note that $\|\tilde{\Psi}^+\|$ only occurs in conjunction with terms involving eigenvalues, such that the overall bound still converges. For example, one can prove that a decay rate of λ_r faster than $O(r^{-12})$ ensures that $E(r, n) = 2r\tilde{a}\|\tilde{\Psi}^+\|\|\tilde{\mathbf{E}}\| \rightarrow 0$ for $r \rightarrow \infty$ independently of n : It holds that

$$E(r, n) = 2r\tilde{a}\|\tilde{\Psi}^+\|\|\tilde{\mathbf{E}}\| = 2r\tilde{a}\left(1 + O(r\lambda_r^{-1/4}n^{-1/4})\right)\left(\Lambda_{\geq r} + O(\sqrt{\Lambda_{> r}}n^{-1/2})\right).$$

If one expands the product, the term which decays slowest with respect to r is (recall that $\tilde{a} = O(\sqrt{r})$)

$$2r\tilde{a}O(r\lambda_r^{-1/4})O(\sqrt{\Lambda_{> r}}) = O(r^{5/2}\lambda_r^{-1/4}\Lambda_{> r}^{1/2}).$$

Now if $\lambda_r = r^{-d}$, then $\Lambda_{> r} = O(r^{1-d})$, and

$$O(r^{5/2}\lambda_r^{-1/4}\Lambda_{> r}^{1/2}) = O(r^{5/2}r^{d/4}r^{(1-d)/2}) = O(r^{3-d/4}).$$

We require that the exponent is smaller than 0 which is true if $d > 12$. Again, since these are worst case considerations, and usually r and n will be coupled in some way, the additive terms will be controlled even for slower decay rates.

An interesting feature of the bound is that it is uniform in i , which means that the bound holds simultaneously for all eigenvectors. Therefore, the individual bounds can be combined, for example, to sums of scalar products without a decrease in the probability with which the bound holds.

In principle, it is possible to further relate the decay rate of the eigenvalues of the kernel matrix l_i to the asymptotic eigenvalue λ_i , for example using bounds for individual eigenvalues (Braun, 2006), or tail-sums of eigenvalues (Blanchard et al., 2007; Shawe-Taylor et al., 2005) if we wish to explicitly control the component of the relevant information vector which is not contained in the leading kernel PCA directions.

Appendix C. A Worked Through Example

In this section, we work through the ‘‘splice’’ data set to show how one would perform a kernel fitness analysis using the methods presented here. The computations of the estimates proposed in Section 4 are summarized in Algorithm 1.

We start out with the splice data set. As explained in the main section, each data points encodes sequence of aminoacids. In the positive examples, there exists a so-called splice site in the center of the encoded DNA signal. The task requires to predict splice sites in these short DNA sequences.

Usually, one would start with some specific kernel, for example an rbf-kernel, train some kernel learning algorithm using this kernel, evaluate the kernel on some test data set, and start to select different parameters. There are two potential drawbacks following this approach: (1) there exists

Algorithm 1 Computing the estimates from Section 4

Input: Kernel matrix \mathbf{K} , label vector Y , loss function L

Output: kernel PCA coefficients z , dimensionality \hat{d} ,
negative log-likelihood $\hat{\ell}$, denoised labels \hat{Y} ,
noise-level $\hat{\epsilon}rr$

```

1: {Compute kernel PCA coefficients}
2: Compute eigendecomposition  $\mathbf{KU} = \mathbf{UD}$ 
3:  $z \leftarrow \mathbf{U}^\top Y$ 
4: {Estimate dimensionality  $\hat{d}$  (Eq. 4)}
5:  $c \leftarrow 0$ ;  $C \leftarrow \|z\|^2$  {here, it is shown in detail how to achieve linear run-time}
6: for  $d = 1$  to  $n/2$  do
7:    $c \leftarrow c + z_d^2$ 
8:    $s_1 \leftarrow c/d$ 
9:    $s_2 \leftarrow (C - c)/(n - d)$ 
10:   $l_d \leftarrow d \log s_1 + (n - d) \log s_2$ 
11: end for
12:  $\hat{d} \leftarrow \operatorname{argmin}_{1 \leq d \leq n/2} l_d$ 
13:  $\hat{\ell} \leftarrow l_{\hat{d}}$ .
14: {Compute denoised labels (Eq. 5)}
15: Extract first  $\hat{d}$  eigenvectors  $\mathbf{T} \leftarrow \mathbf{U}_{:,1:\hat{d}}$ 
16:  $\hat{Y} \leftarrow \mathbf{T}\mathbf{T}^\top Y$ 
17: {Estimate noise-level (Eq. 6)}
18:  $\hat{\epsilon}rr = \frac{1}{n} \sum_{i=1}^n L(Y, \hat{Y})$ 

```

no absolute measure of the goodness of a certain kernel choice, only comparisons to other kernels, (2) there exists some dependency on the kernel learning method employed. Using the methods developed in this paper, it is possible to explore the relationship between the kernel and the data set in an algorithm independent way. Furthermore, in the case of poor performance, it is possible to distinguish between very complex cases (which require more input data), and cases where the data set appears to be very noisy (either requiring better data quality, or a kernel which can capture more information about the learning problem).

The splice data set consists of 20 resamples. We first try an rbf-kernel with width $w = 50$ (see Section A). We start by computing and plotting the kernel PCA coefficients. The resulting coefficients are plotted in Figure 9(a). We see that the data set appears to be rather high-dimensional, and the noise level is also quite high. The estimated median estimated dimension is 87.5, but it seems that roughly up to dimension 200, relevant information might be contained.

As explained in the main text, the encoding used by the rbf-kernel is not fit for this example. The four aminoacids A, C, G, and T have just been mapped to the numbers 1–4. We re-encode the object features by mapping A, C, G, and T to the four vectors $(1, 0, 0, 0)$, $(0, 1, 0, 0)$, and so on. The resulting kernel PCA coefficients are plotted in Figure 9(b). The encoding has obviously resulted in a large improvement, as the dimension is much smaller now, while the amount of noise has also been reduced.

Finally, we consider using a weighted-degree-kernel (Sonnenburg et al., 2005). The resulting kernel PCA coefficients are plotted in Figure 9(c). While the estimated dimension is larger than

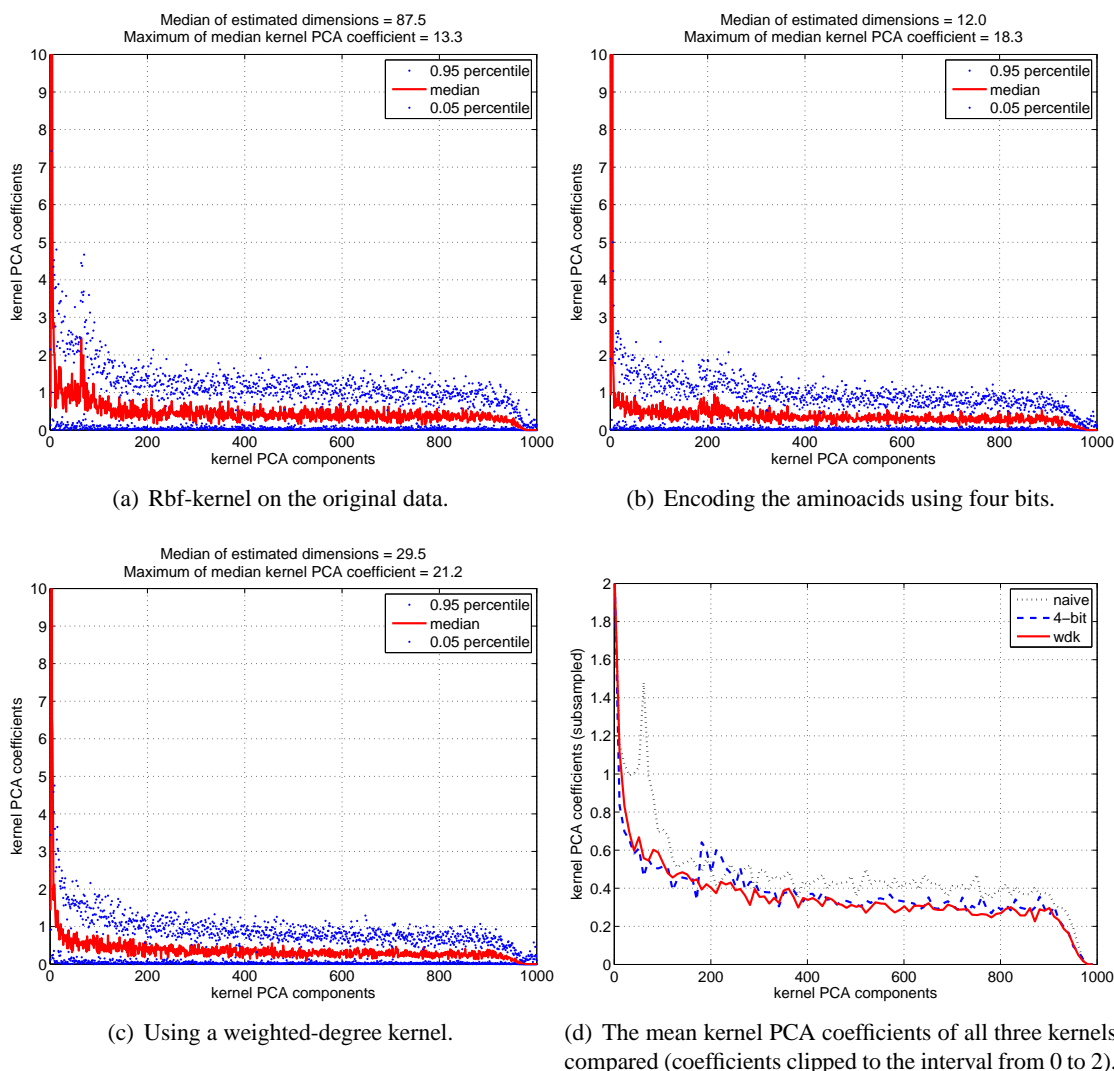


Figure 9: Figures (a)-(c) show 0.05, 0.5, and 0.95 percentiles of the kernel PCA coefficients over the 20 resamples of the *splice* data set using the indicated kernels. Coefficients have been truncated to the range $[0, 10]$ for better visibility. Figure (d) plots all three medians for comparison (subsampled by combining ten consecutive points into their mean for better visibility). Coefficients are sorted by decreasing corresponding eigenvalue.

in the previous case, the amount of noise was dramatically reduced, which is also reflected in the classification results shown in Table 5.

In summary, using the estimates here, one can get a much more fine-grained assessment of how well a kernel is adapted to the data. Figure 9(d) compares the mean kernel PCA coefficients over the resamples for the three kernels. Initially, the *splice* data set appears to be rather high-dimensional, indicating that more data would be needed. Incorporating domain knowledge in the encoding and finally switching to a special-purpose kernel shows that the true dimensionality of the data is in fact

smaller, and that the noise level, which was initially quite high, could also be lowered significantly. Using the weighted-degree-kernel the data quality and the amount of data seem to be suited for predicting with high accuracy.

Appendix D. A Sin-Theta-Theorem

The following theorem is a special case of Theorem 6.2 in the book by Davis and Kahan (1970).

Theorem 2 *Let \mathbf{A} be a symmetric $n \times n$ -matrix with eigendecomposition $\mathbf{A} = \mathbf{U}\mathbf{L}\mathbf{U}^\top$. Let \mathbf{U} and \mathbf{L} be partitioned as follows:*

$$\mathbf{U} = [\mathbf{U}_1 \ \mathbf{U}_2], \quad \mathbf{L} = \begin{bmatrix} \mathbf{L}_1 & 0 \\ 0 & \mathbf{L}_2 \end{bmatrix},$$

where \mathbf{U}_1 is an $n \times k$ -matrix, \mathbf{L}_1 is a $k \times k$ -matrix, \mathbf{U}_2 is an $n \times n - k$ -matrix, and \mathbf{L}_2 is an $n - k \times n - k$ -matrix. Furthermore, let \mathbf{E} be another symmetric $n \times n$ -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,

$$\|\mathbf{U}_2^\top \tilde{x}\| \leq \frac{\|\mathbf{E}\|}{\min_{n-k \leq i \leq n} |\tilde{l} - l_i|}.$$

The proof of this theorem can also be found in the thesis of Braun (2005), Lemma 4.50, p. 70.

References

- Gilles Blanchard, Olivier Bousquet, and Laurent Zwald. Statistical properties of kernel principal component analysis. *Machine Learning*, 66(2–3):259–294, 2007.
- Mikio L. Braun. Accurate error bounds for the eigenvalues of the kernel matrix. *Journal of Machine Learning Research*, 7:2303–2328, Nov 2006.
- Mikio L. Braun. *Spectral Properties of the Kernel Matrix and Their Application to Kernel Methods in Machine Learning*. PhD thesis, University of Bonn, 2005. Available electronically at http://hss.ulb.uni-bonn.de/diss_online/math_nat_fak/2005/braun_mikio.
- Chris J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- Chandler Davis and William M. Kahan. The rotation of eigenvectors by a perturbation, iii. *SIAM Journal of Numerical Analysis*, 7:1–46, 1970.
- Theodoros Evgeniou and Massimiliano Pontil. On the V_γ dimension for regression in reproducing kernel hilbert spaces. In *Proceedings of Algorithmic Learning Theory*, 1999.
- Edwin T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 160(620–630), 1957.
- Vladimir Koltchinskii and Evariste Giné. Random matrix approximation of spectra of integral operators. *Bernoulli*, 6(1):113–167, 2000.

- Vladimir I. Koltchinskii. Asymptotics of spectral projections of some random matrices approximating integral operators. *Progress in Probability*, 43:191–227, 1998.
- Sebastian Mika. *Kernel Fisher Discriminants*. PhD thesis, Technische Universität Berlin, December 2002.
- Klaus-Robert Müller, Sebastian Mika, Gunnar Rätsch, Koji Tsuda, and Bernhard Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transaction on Neural Networks*, 12(2):181–201, May 2001.
- Gunnar Rätsch, Takashi Onoda, and Klaus-Robert Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, March 2001.
- Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. MIT Press, 2002.
- Bernhard Schölkopf, Alexander J. Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
- Bernhard Schölkopf, Sebastian Mika, Christopher J. C. Burges, Philipp Knirsch, Klaus-Robert Müller, Gunnar Rätsch, and Alex J. Smola. Input space vs. feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, 1999.
- John Shawe-Taylor, Peter L. Bartlett, Robert C. Williamson, and Martin Anthony. Structural risk minimization over date-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.
- John Shawe-Taylor, Christopher K. I. Williams, Nello Christianini, and Jaz Kandola. On the eigen-spectrum of the Gram matrix and the generalization error of kernel-PCA. *IEEE Transactions on Information Theory*, 51(7):2510–2522, July 2005.
- Alex J. Smola, Bernhard Schölkopf, and Klaus-Robert Müller. The connection between regularization operators and support vector kernels. *Neural Networks*, 11(4):637–649, 1998.
- Sören Sonnenburg, Gunnar Rätsch, and Bernhard Schölkopf. Large scale genomic sequence SVM classifiers. In *Proceedings of the 22nd International Machine Learning Conference*, pages 848–855. ACM Press, 2005.
- Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.
- Vladimir Vapnik. *Statistical Learning Theory*. Wiley, 1998.
- Régis Vert, Laurent Zwald, Gilles Blanchard, and Pascal Massart. Kernel projection machine: a new tool for pattern recognition. In *Advances in Neural Information Processing Systems (NIPS 2004)*, pages 1649–1656. 2005, 2005.
- Ulrike von Luxburg. *Statistical Learning with Similarity and Dissimilarity Functions*. PhD thesis, Technische Universität Berlin, 2004.
- Grace Wahba. *Spline Models For Observational Data*. Society for Industrial and Applied Mathematics, 1990.

Robert C. Williamson, Alex J. Smola, and Bernhard Schölkopf. Generalization bounds for regularization networks and support vector machines via entropy numbers of compact operators. *IEEE Transaction on Information Theory*, 47(6):2516–2532, 2001.

Tong Zhang. Learning bounds for kernel regression using effective data dimensionality. *Neural Computation*, 17:2077–2098, 2005.

Laurent Zwald and Gilles Blanchard. On the convergence of eigenspaces in kernel principal components analysis. In *Advances in Neural Information Processing Systems (NIPS 2005)*, volume 18, 2006.