KERNEL PCA WITH THE NYSTRÖM METHOD

FREDRIK HALLGREN

Department of Statistical Science
University College London

ABSTRACT. Kernel methods are powerful but computationally demanding techniques for non-linear learning. A popular remedy, the Nyström method has been shown to be able to scale up kernel methods to very large datasets with little loss in accuracy. However, kernel PCA with the Nyström method has not been widely studied. In this paper we derive kernel PCA with the Nyström method and study its accuracy, providing a finite-sample confidence bound on the difference between the Nyström and standard empirical reconstruction errors. The behaviours of the method and bound are illustrated through extensive computer experiments on real-world data. As an application of the method we present kernel principal component regression with the Nyström method.

Keywords: Kernel methods, non-parametric statistics, confidence interval, dimensionality reduction, unsupervised learning, learning theory, PCA, functional PCA, PCR, MDS

CONTENTS

E-mail address: `fredrik.hallgren@ucl.ac.uk`.

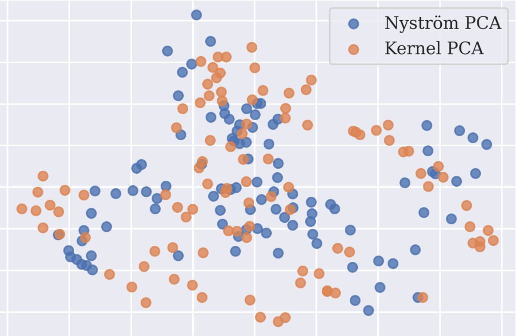
1. Introduction

Kernel methods generalize classical statistical methods to discover non-linear patterns in data [Shawe-Taylor and Cristianini, 2004]. They have been demonstrated to achieve state-of-the-art results in many application domains and it is straightforward to apply them to non-numeric data, such as graphs or text [Vishwanathan et al., 2010, Lodhi et al., 2002]. Through a near arbitrary non-linear mapping of data points into a Hilbert space they offer remarkable flexibility whilst providing a precise mathematical framework for statistical analyses. A host of linear statistical methods have been adapted to be used with kernels, including Fisher discriminant analysis (FDA) [Mika et al., 1999], independent component analysis (ICA) [Bach and Jordan, 2002], instrumental variable (IV) regression [Singh et al., 2019], and many more. Kernel PCA is a non-linear version of principal component analysis (PCA), a ubiquitous method to discover the most important directions of variation in data [Pearson, 1901]. PCA may be used for dimensionality reduction, exploratory data analysis, anomaly detection, discriminant analysis, clustering, or as a general preprocessing step for regression or classification [Jolliffe, 2002, Wold et al., 1987].

The other side of the coin of kernel methods is their large computational requirements, as they generally scale in the number of data points rather than the number of data dimensions. As a remedy, various approximations have been proposed, such as the Nyström method, which randomly selects a smaller subset of data points and looks for solutions in their linear span. The Nyström method also plays an important role in recent state-of-the-art implementations of kernel methods [Rudi et al., 2017, Ma and Belkin, 2017, Meanti et al., 2020, Carratino et al., 2021].

The need for approximate methods becomes particularly acute for kernel PCA, since it relies on the eigendecomposition of the kernel matrix, which requires about $9n^3 + \mathcal{O}(n^2)$ floating-point operations, as opposed to $\frac{1}{3}n^3 + \mathcal{O}(n^2)$ floating-point operations for the solution of a linear system by way of the Cholesky decomposition when performing regression [Golub and Van Loan, 2013, Chapters 4, 8]. Despite this fact, kernel PCA with the Nyström method has not yet been derived fully in line with linear PCA.

In this paper we derive kernel PCA with the Nyström method, generalizing a previous method from Sterge et al. [2020]. We provide orthonormal principal components in the span of the Nyström subset that maximize the variance of the data, without assuming that the data has zero mean, as well as the associated principal scores[1]. The principal scores are perhaps of particular interest, since they allow for the method to be used as a preprocessing step before applying supervised learning methods, by virtue of providing a new representation of data points in the new coordinate system defined by the principal components. The figure to the right shows the first two dimensions for these representations for an example with a dataset of images of handwritten digits, in comparison with standard full kernel PCA[2].



The principal scores are given as follows. First let K_{mm} be m randomly subsampled rows and columns of the original kernel matrix K, and K_{nm} be the same m subsampled columns. Centring

[1] Different conventions exist for the terminology of PCA. Throughout this paper we will take the *principal components* to mean the vectors defining the subspaces that maximize the variance of the data i.e. the eigenvectors of the centred covariance operator or matrix. These are elsewhere sometimes referred to as the *principal axes*.

[2] Please see `https://github.com/fredhallgren/nystrompca` for details

the data in feature space corresponds to adjusting these matrices through

$$K'_{nm} = K_{nm} - \mathbb{1}_n K_{nm} - \widetilde{K}\,\mathbb{1}_n^{n,m} + \mathbb{1}_n \widetilde{K}\,\mathbb{1}_n^{n,m}$$
$$K'_{mm} = K_{mm} - \mathbb{1}_n^{m,n} K_{nm} - K_{mn}\mathbb{1}_n^{n,m} + \mathbb{1}_n^{m,n}\widetilde{K}\,\mathbb{1}_n^{n,m}$$

where $\widetilde{K} = K_{nm}K_{mm}^{-1}K_{mn}$ and $\mathbb{1}_n$, $\mathbb{1}_n^{n,m}$ and $\mathbb{1}_n^{m,n}$ are $n\times n$, $n\times m$ and $m\times n$ matrices respectively with all elements equal to $\frac{1}{n}$. Each element of $\mathbb{1}_n K_{nm}$ or $\mathbb{1}_n^{m,n}K_{nm}$ equals the mean of the values in K_{nm} in that element's column, and each element of $\widetilde{K}\mathbb{1}_n^{n,m}$ or $K_{mn}\mathbb{1}_n^{n,m}$ equals the mean across that row. The matrices $\mathbb{1}_n^{m,n}\widetilde{K}\mathbb{1}_n^{n,m}$ and $\mathbb{1}_n\widetilde{K}\mathbb{1}_n^{n,m}$ are constant with each element equal to the sum of all elements of $\widetilde{K}$ divided by n^2. Now create an approximate kernel matrix through

$$\frac{1}{n}\widetilde{K}' = \frac{1}{n}K'^{-1/2}_{mm}K'_{mn}K'_{nm}K'^{-1/2}_{mm}$$

and calculate its eigenvalues $\widetilde{\lambda}_j$ and eigendecomposition $V\widetilde{\Lambda}V^T$, where $M^{-1/2} = UD^{-1/2}U^T$ for a matrix M with eigendecomposition UDU^T. The scores are then given by $W = K'_{nm}K'^{-1/2}_{mm}V$, which is a new data matrix with observations along the rows, and the variances of the new data variables (in the columns) are given by $\widetilde{\lambda}_j$. The method has time complexity $\mathcal{O}(nm^2)$ which is the same as when the Nyström method is applied to regression.

The method centres the data in the feature space, as is the case for linear PCA [Jolliffe and Cadima, 2016]. Without this adjustment, the lines defined by the principal components along which the variance is maximized are forced to go through the origin, no longer maximizing the variance in an unconstrained manner and requiring an assumption of zero-mean data in feature space.

We further study the statistical accuracy of the proposed method. In the special case when the number of subsampled data points for the Nyström method equals the PCA dimension, then both the empirical and true reconstruction errors of the Nyström method equal the corresponding reconstruction errors for kernel PCA constructed using only the subset of data points. For the general case we provide a finite-sample confidence bound (a confidence interval) with $\mathcal{O}(m^3)$ time complexity that doesn't require that we have observed the entire dataset, only the subset of data [Ramachandran and Tsokos, 2015]. In line with most results on the accuracy of kernel PCA we here assume that data has zero mean. The result states that with high probability, the difference between the empirical reconstruction errors of Nyström kernel PCA and full kernel PCA is less than or equal to a data-dependent quantity

$$R_n(\widetilde{V}_d) - R_n(\hat{V}_d) \le h\left(\sup_x k(x,x),\ \left\{\hat{\lambda}_j\right\}_{j=1}^{d+1},\ m,\ n\right)$$

which depends on the maximum value of the kernel function, the eigenvalues of the kernel matrix from the subset of randomly subsampled data points, the size of this subset m and the total size of the dataset n, where $h(\cdot)$ is a fixed function and $d < m$. Please see Section 6 for the complete result.

We illustrate and evaluate the proposed method and derived confidence bound through experimental analysis using several different datasets and kernel functions. We first compare the accuracy of Nyström kernel PCA with a number of other unsupervised learning methods, where its performance is seen to be very close to full kernel PCA, whilst being much more efficient. Then we illustrate the behaviour of the bound across different PCA dimensions. The source code for all the experiments is publicly available at `https://github.com/fredhallgren/nystrompca`.

From the proof of the confidence bound one can deduce sharper versions of some concentration results from Rosasco et al. [2010] on the empirical covariance operator and its eigenvalues. Please see Section 6.1 for details.

To demonstrate the use of Nyström kernel PCA with supervised learning methods we apply it to the regression problem to present kernel principal component regression with the Nyström method. Principal components regression (PCR) performs a linear regression on the principal scores from the top principal components instead of the original data and introduces regularization for improved generalization. We also illustrate the method through experimental analysis and compare it to kernel ridge regression with the Nyström method. In summary, the prediction for a data point x^* is given by

$$\hat{y} = \bar{y} + y'^T K'_{nm} K'^{-1/2}_{mm} V_d \widetilde{\Lambda}_d^{-1} V_d^T K'^{-1/2}_{mm} \widetilde{\kappa}(x^*)$$

where $y' = (y_1 - \bar{y},\ y_2 - \bar{y},\ ...,\ y_n - \bar{y})^T$ and $\widetilde{\kappa}(x) = \kappa_m(x) - K_{mn}\mathbf{1}_n - \mathbb{1}_n^{m,n} K_{nm} K_{mm}^{-1} \kappa_m(x) + \mathbb{1}_n^{m,n} \widetilde{K} \mathbf{1}_n$ with $\mathbf{1}_n$ a length-n column vector given by $\mathbf{1}_n = (\frac{1}{n}, \frac{1}{n},\ ...,\ \frac{1}{n})^T$ and $\kappa_m(x) = (k(x_1, x),\ k(x_2, x),\ ...,\ k(x_m, x))^T$. Using similar techniques we also present a novel derivation of standard kernel PCR with centred data in feature space, where a prediction is given by

$$\hat{y} = \bar{y} + y'^T Q_d \Lambda_d^{-1} Q_d^T \kappa'(x^*)$$

where $Q_d \Lambda_d Q_d^T$ is the truncated eigendecomposition of $K' = K - \mathbb{1}_n K - K \mathbb{1}_n + \mathbb{1}_n K \mathbb{1}_n$ and $\kappa'(x) = \kappa(x) - \mathbb{1}_n \kappa(x) - K\mathbf{1}_n + \mathbb{1}_n K \mathbf{1}_n$ with $\kappa(x) = (k(x_1, x), k(x_2, x), ..., k(x_n, x))^T$.

A summary of our main contributions is as follows

(1) Deriving kernel PCA with the Nyström method

(2) A result on the accuracy in the special case of $d = m$ for both the empirical and true errors

(3) A finite-sample confidence bound for the empirical error in the general case

(4) Presenting kernel principal component regression with the Nyström method

(5) Novel specification of kernel PCR with centred regressors

(6) Sharper versions of some concentration results from previous literature

In the next section we give an overview of previous work (Section 2), then go through relevant background (Section 3), present the main method (Section 4), study the special case when $d = m$ (Section 5), provide the confidence bound on the accuracy of the method (Section 6), conduct experimental analysis of the method and bound (Section 7), present kernel principal component regression with the Nyström method (Section 8) and finally conclude with a summary and outlook (Section 9). Proofs are in the appendix.

Notation. Upper-case letters will be used for matrices and operators and generally for random variables, unless they represent data points before they are observed. Vectors in $\mathbb{R}^p$ will be denoted by small letters and parameters fitted to data often by letters from the Greek alphabet. A row vector v in $\mathbb{R}^p$ with elements $v_1, v_2, ..., v_p$ will be written $(v_1, v_2, ..., v_p)$. The transpose of a vector or matrix is v^T. If not stated otherwise all Euclidean vectors will be column vectors. The arithmetic mean of a vector is denoted $\bar{v}$. Indices for data points will be denoted by i, r, or ℓ; indices for eigenvectors or dimensions will be denoted by j, k, p or q. Estimated quantities will often be denoted by $\hat{\cdot}$, approximations by $\tilde{\cdot}$ and centred quantities by $\cdot'$. Empirical quantities may be superscripted or subscripted by the number of observations used in the estimate. The probability density function of a measure $\mathbb{P}_Y$ will be denoted by $p_Y(y)$. The symbol Y will be used for a generic random variable; the symbols T or L for a generic operator and the symbol M for a generic matrix. The linear span of a set of vectors A is written $\mathrm{span}\{A\}$ or $\langle A\rangle$. The cardinality of a basis for the space V is written $\dim(V)$.

The symbol $\mathcal{O}(\cdot)$ denotes Big-O notation [Sipser, 2013]. The function $\lambda_j(\cdot)$ returns the jth eigenvalue, in decreasing order, of its argument, and the symbol $\lambda_{<d}$ denotes the sum of the largest d eigenvalues $\lambda_1, \lambda_2, ..., \lambda_d$. If v is majorized by u we write $v \succ u$. The symbol $:=$ denotes the introduction of new notation, i.e. $a := b$ means that b will be denoted by a, and vice versa for $a =: b$. The binary operators $\vee$ and $\wedge$ are defined as $a \vee b = \max\{a, b\}$ and $a \wedge b = \min\{a, b\}$. The notation $\otimes^n v$ is used for $v \otimes v \otimes \cdots \otimes v$ for n instances of the symbol v.

The functional $\|\cdot\|$ denotes the operator norm or the Euclidean norm, depending on the context. For other norms the space will always be specified. For a Banach space $\mathcal{B}$, we let $\mathcal{B}^*$ denote the dual space of bounded linear functionals on $\mathcal{B}$. For an operator T, we let T^* denote its adjoint. The image of an operator is $\mathrm{Im}(T)$ and its null space (also called its kernel) is $\mathrm{Ker}(T)$.

2. Previous work

The study of the statistical accuracy of kernel PCA, or of the related problems of functional PCA [Besse and Ramsay, 1986, Hall et al., 2006] and PCA of a Hilbert space-valued random variable [Besse, 1991], was initiated in Dauxois et al. [1982]. They demonstrated the consistency of the reconstruction error and asymptotic normality of the empirical reconstruction error and principal components about the true quantities. The asymptotics of kernel PCA was also studied in Koltchinskii and Giné [2000]. A concentration inequality for the empirical reconstruction error versus its expectation, based on McDiarmid's inequality [McDiarmid, 1989], was provided in Shawe-Taylor et al. [2002] and the same authors later presented a confidence bound on the expected empirical reconstruction error versus the true error [Shawe-Taylor et al., 2005]. In this bound the expectation is with respect to the data point to be projected and the confidence with respect to different training datasets. A similar bound, as well as a version for centred kernel PCA, was later presented in Blanchard et al. [2007]. The centred version is more conservative compared to the uncentred one. Approximate confidence bounds for both the principal values and components were given in Hall and Hosseini-Nasab [2006] based on the bootstrap method [Davison and Hinkley, 1997]. However, these results are not immediately applicable to kernel PCA since the kernel is defined on a compact subset of $\mathbb{R} \times \mathbb{R}$. The current state-of-the-art for empirically measuring the accuracy of kernel PCA appears to be Haddouche et al. [2020].

The Nyström method has been widely studied for different settings and assumptions. Originally developed for the discretization of integral equations [Nyström, 1930, Banach, 1932], it was adapted to kernel methods in Williams and Seeger [2001] and applied to regression. The accuracy of the approximate kernel matrix versus the full kernel matrix, considering the full dataset as fixed, has been studied in a number of papers, please see Gittens and Mahoney [2016] and references therein. The study of the accuracy of the Nyström method as applied to regression culminated in the seminal work by Rudi et al. [2015] as a probabilistic bound on the expected regression error with general assumptions.

A recent paper [Sterge et al., 2020] presented a similar method for kernel PCA with the Nyström method, but under an assumption of zero-mean data in feature space, and in the current work we also derive the principal scores and provide an empirical evaluation of our method. They also presented a probabilistic inequality for the true reconstruction error with respect to the empirical subspace, which depends on the maximum value of the kernel function $\sup_x k(x,x)$, the total number of data points n and the covariance operator C from the true distribution $\mathbb{P}$. As a corollary they also presented an asymtoptic rate of convergence under the assumption of polynomial decay of the eigenvalues of C. Even more recently Sterge and Sriperumbudur [2021] presented a similar analysis to [Sterge et al., 2020], but with one way of centring the data. This method is different from the one considered here, with the top principal component given by

$$\tilde{\phi}_1 = \sqrt{m} G_m^* K_{mm}^{-1/2} u_1$$

where u_1 is the top eigenvector of $\frac{1}{n-1} K_{mm}^{-1/2}(K_{mn} - K_{mn}\mathbb{1}_n)K_{nm}K_{mm}^{-1/2}$ and G_m^* is given by $\alpha \mapsto \frac{1}{\sqrt{m}} \sum_{k=1}^m \alpha_k k(x_k, x)$.

If we assume data to have zero mean in feature space, then our derived principal scores are somewhat similar to the *virtual samples* of Golts and Elad [2016] which were introduced in the context of dictionary learning [Aharon et al., 2006] (also see Vincent and Bengio [2002], Guigue et al. [2005]). These may also be used as a drop-in replacement for the original data points, but they are not uncorrelated and don't correspond to the principal scores.

Another related method was described in [Iosifidis and Gabbouj, 2016]. They first derive low-rank data representations from the uncentred Nyström kernel matrix, similar to the *virtual samples* above, including for novel unseen data points. They also propose a method to centre the data in the feature space, although this is done in order to make the data representations into a subspace and the centred representations are different from the principal scores derived below. The centring of the matrices K_{nm} and K_{mm} is the same as the one used here, but these matrices are applied differently.

3. Background

We have a reproducing kernel Hilbert space $\mathcal{H}$ (RKHS) of functions from a set $\mathcal{X}$ to the real numbers. Associated with each RKHS is a symmetric positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with a reproducing property $\langle k(x,\cdot), f\rangle_{\mathcal{H}} = f(x)$ for which the point evaluation $f \mapsto \langle k(x,\cdot), f\rangle_{\mathcal{H}}$ is bounded. The kernel maps each element $x \in \mathcal{X}$ to an element $\phi(x) := k(x,\cdot) \in \mathcal{H}$. We assume $\mathcal{H}$ is separable, which will be the case for example if k is continuous and $\mathcal{X}$ is compact [Paulsen and Raghupathi, 2016].

We have observations $\{x_i\}_{i=1}^n$ of an $\mathcal{X}$-valued random variable $X : (\Omega, \mathcal{A}, \mathbb{P}) \to (\mathcal{X}, \mathcal{A}_{\mathcal{X}}, \mathbb{P}_X)$ where $\mathbb{P}_X(A) = \mathbb{P}(X^{-1}(A))$ [Cohn, 1980, Graham and Talay, 2011]. We assume X is absolutely continuous and that it has a continuous density and so all x_i will be distinct. We obtain a random variable $Z = \phi(X) \in \mathcal{H}$ with observations $z_i = \phi(x_i)$, assuming that ϕ is measurable, which will be the case for example when k is continuous. Its expectation in $\mathcal{H}$ is given by $\mathbb{E}[Z] = \int Z d\mathbb{P}$. We assume Z is in $L^1(\Omega, \mathcal{A}, \mathbb{P}; \mathcal{H})$ with norm $\mathbb{E}[\|Z\|_{\mathcal{H}}] = \int \|Z\|_{\mathcal{H}} d\mathbb{P}$ and so also is square-integrable [Ledoux and Talagrand, 2013].

Observation of an integrable random variable Y with values in some Banach space $\mathcal{B}$ (such as $\mathcal{H}$, or $\mathbb{R}$ with norm $\|\cdot\|_{\mathcal{B}} = |\cdot|$) and with observations $y_1, y_2, ..., y_M$ corresponds to application of the evaluation operator $E_\omega : L^1(\Omega, \mathcal{A}, \mathbb{P}_{\mathcal{B}}; \mathcal{B}) \to \mathcal{B}$,

$$E_\omega(Y) = Y(\omega)$$

or for specific data points

$$E_{\omega_i}(Y) = Y(\omega_i) = y_i$$

The evaluation operator is linear, since $E_\omega(aY_1 + bY_2) = (aY_1 + bY_2)(\omega) = aY_1(\omega) + bY_2(\omega)$ for $a, b \in \mathbb{R}$ with norm

$$\|E_\omega\| = \sup_{\|Y\|_1=1} \|E_\omega(Y)\|_{\mathcal{B}} = \sup_{\|Y\|_1=1} \|Y(\omega)\|_{\mathcal{B}} = \sup_{y\in\mathcal{B}} \|y\|_{\mathcal{B}}$$

Principal component analysis (PCA) of the zero-mean random variable $Z \in \mathcal{H}$ constructs an optimal subspace $V_d \subset \mathcal{H}$, of dimension d, such that the so-called reconstruction error

$$R(V) = \mathbb{E}\left[\|P_V Z - Z\|^2_{\mathcal{H}}\right]$$

is minimized, where $P_V : \mathcal{H} \to \mathcal{H}$ is the projection of (a realization of) Z on a subspace V [Besse, 1991]. This is termed the *true* reconstruction error [Blanchard et al., 2007]. Since Z is square-integrable the reconstruction error always exists and is finite.

In other words, the optimal d-dimensional subspace V_d is given by

$$V_d = \underset{\dim(V)=d}{\arg\min}\ \mathbb{E}\left[\|P_V Z - Z\|^2_{\mathcal{H}}\right]$$

An estimate of the optimal subspace V_d is obtained from the data $\{z_i\}_{i=1}^n$ by minimizing the *empirical* reconstruction error

$$R_n(V) = \frac{1}{n}\sum_{i=1}^n \|P_V z_i - z_i\|^2_{\mathcal{H}}$$

which has a unique minimum since all eigenvalues are distinct [Blanchard et al., 2007]. We denote the estimated subspace by $\hat{V}_d$. One may also consider the true reconstruction error with respect to the empirical subspace, given by

$$R(\hat{V}_d) = \mathbb{E}\left[\|P_{\hat{V}_d} Z - Z\|^2_{\mathcal{H}}\right]$$

where the expectation may be taken both with respect to Z and $\hat{V}_d$, or treating the subspace as fixed; as well as the expected value of the empirical reconstruction error, given by

$$\mathbb{E}\left[R_n(\hat{V}_d)\right] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \|P_{\hat{V}_d} z_i - z_i\|^2_{\mathcal{H}}\right]$$

When the random variable Z is not assumed to have zero mean, the smallest reconstruction error is obtained from the centred random variable $Z' = Z - \mathbb{E}[Z]$

$$R(V_d) = \min_{\dim(V)=d} \mathbb{E}[\|P_V Z' - Z'\|_{\mathcal{H}}^2]$$

and similarly for the empirical reconstruction error replacing z_i by $z_i' = z_i - \frac{1}{n}\sum_{\ell=1}^n z_\ell$.

Alternatively, instead of minimizing the reconstruction error over d-dimensional subspaces V using the centred random variable, one may minimize over affine subspaces with respect to the original random variable, and also optimize with respect to the term used for centring

$$R(V_d) = \min_{\substack{a\in\mathcal{H}\\ \dim(V)=d}} \mathbb{E}[\|P_{a+V}Z - Z\|_{\mathcal{H}}^2] = \min_{\substack{a\in\mathcal{H}\\ \dim(V)=d}} \mathbb{E}[\|P_V(Z-a) - (Z-a)\|_{\mathcal{H}}^2]$$

where a is the translation of the vector space V, and whose optimal value is known to equal $\mathbb{E}[Z]$, and $P_{a+V}Z = a + P_V(Z-a)$ is the affine projection.

The *covariance operator* is an element $C(u,v) \in \mathcal{H}\otimes\mathcal{H}$ in the tensor product of bilinear functionals on $\mathcal{H}$, given by $C(u,v) = \mathbb{E}[Z\otimes Z]$. The *centred* covariance operator is given by

$$C'(u,v) = \mathbb{E}[(Z-\mathbb{E}[Z])\otimes(Z-\mathbb{E}[Z])] = \mathbb{E}[Z'\otimes Z']$$

Identifying $\mathcal{H}\otimes\mathcal{H}$ with the space $\mathrm{HS}(\mathcal{H})$ of Hilbert-Schmidt operators on $\mathcal{H}$ by way of the mapping of elementary tensors $u\otimes v \mapsto \langle\cdot, u\rangle_{\mathcal{H}} v$ we obtain $C' = \mathbb{E}[\langle\,\cdot\,, Z'\rangle_{\mathcal{H}} Z']$. When we refer to the covariance operator we may either refer to the tensor in $\mathcal{H}\otimes\mathcal{H}$ or the operator in $\mathrm{HS}(\mathcal{H})$.

A Hilbert-Schmidt operator L is an operator on a Hilbert space $\mathcal{H}$ with finite Hilbert-Schmidt norm, given by $\|L\|_{\mathrm{HS}(\mathcal{H})} = \sum_i \|Le_i\|_{\mathcal{H}}$ for any orthonormal basis $\{e_i\}_i$ in $\mathcal{H}$ [Davies, 2007, Chapter 5]. It is a Hilbert space, with inner product $\langle L_1, L_2\rangle_{\mathrm{HS}(\mathcal{H})} = \sum_i \langle L_1 e_i, L_2 e_i\rangle_{\mathcal{H}}$. The Hilbert-Schmidt norm is always larger than or equal to the operator norm, $\|L\| \le \|L\|_{\mathrm{HS}(\mathcal{H})}$, and if $\mathcal{H}$ is finite it coincides with the Frobenius norm, $\|L\|_{\mathrm{HS}(\mathcal{H})} = \|M\|_F$ where M is a matrix representation of L [Kreyszig, 1989].

The covariance operator C' is compact, since it is Hilbert-Schmidt, and so its spectrum is countable and all spectral values are eigenvalues apart from possibly 0. Since C' is infinite-dimensional, by assumption, the value 0 is always a spectral value. Furthermore, the covariance operator is self-adjoint, and so the spectrum is real and the resolvent spectrum is empty. Finally, it is positive and so the spectrum is positive.

The sum of the smallest eigenvalues of the centred operator C' equal the minimum true reconstruction error of the centred random variable $Z' = Z - \mathbb{E}[Z]$. The eigenvectors form a countable orthonormal basis of $\mathrm{Im}(C')$, which can be extended to a countable orthonormal basis for the entire space, since $\mathcal{H}$ is separable. Denoting the eigenvalues by $\{\lambda_i\}_{i=1}^\infty$ in decreasing order the minimum reconstruction error can be written $R(V_d) = \sum_{i=d+1}^\infty \lambda_i$.

Replacing the measure $\mathbb{P}_Z$ on $\mathcal{H}$ by the empirical measure $\mathbb{P}_n = \frac{1}{n}\sum_{i=1}^n \delta_{z_i}$, where δ_x is the Dirac delta function, we obtain the empirical covariance operator $C_n' : \mathcal{H}\to\mathcal{H}$

$$C_n' = \frac{1}{n}\sum_{i=1}^n \langle\,\cdot\,, z_i'\,\rangle_{\mathcal{H}} z_i'$$

We denote its eigenvalues by $\hat{\lambda}_1^n, \hat{\lambda}_2^n, ..., \hat{\lambda}_n^n$ in decreasing order and the corresponding eigenvectors by $\hat{\phi}_1^n, \hat{\phi}_2^n, ..., \hat{\phi}_n^n$. It has finite rank, and so the spectrum only contains eigenvalues, and may or may not include 0. The minimum empirical reconstruction error is given by its smallest eigenvalues, $R_n(\hat{V}_d) = \sum_{i=d+1}^n \hat{\lambda}_i^n$, and it can be decomposed as $C_n' = \sum_{i=1}^n \hat{\lambda}_i^n \langle \cdot, \hat{\phi}_i^n \rangle_{\mathcal{H}} \hat{\phi}_i^n$.

If $s : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is square-integrable in the second variable, then the operator given by

$$T_s f = \int_{\mathcal{X}} s(x, y) f(y) d\mathbb{P}_X(y)$$

is an isometry of $L^2(\mathcal{X}, \mathcal{A}_{\mathcal{X}}, \mathbb{P}_X; \mathbb{R})$ into the RKHS with kernel $k(x, y) = \int_{\mathcal{X}} s(x, z)s(z, y)d\mathbb{P}_X(z)$ [Paulsen and Raghupathi, 2016]. One may also consider the integral operator

$$T_k f = \int_{\mathcal{X}} k(x, y) f(y) d\mathbb{P}_X(y)$$

which is equal to $T_k = T_s^2$ and whose eigenvalues equal those of the covariance operator C [Bach, 2017, Shawe-Taylor et al., 2005].

If one replaces the probability measure $\mathbb{P}_X$ by its empirical equivalent $\mathbb{P}_n = \frac{1}{n} \sum_{i=1}^n \delta_{x_i}$with respect to the data points $\{x_i\}_{i=1}^n$ one again obtains an empirical operator T_n

$$T_n f = \int_{\mathcal{X}} k(x, y) f(y) d\mathbb{P}_n(y) = \frac{1}{n} \sum_{i=1}^n k(x, x_i) f(x_i)$$

The sampling operator $G_n, f \mapsto \frac{1}{\sqrt{n}}(f(x_1), f(x_2), ..., f(x_n)$ defines an isometry of $L^2(\mathcal{X}, \mathcal{A}_{\mathcal{X}}, \mathbb{P}_X; \mathbb{R})$ into $\mathbb{R}^n$ which identifies T_n with K [Koltchinskii and Giné, 2000]. Its adjoint G_n^* is given by $\alpha \mapsto \frac{1}{\sqrt{n}} \sum_{k=1}^n \alpha_k k(x_k, x)$ [Rudi et al., 2015]. Furthermore, $C_n = G_n^* G_n$ and $\frac{1}{n}K = G_n G_n^*$.

And so the eigenvalues of the empirical kernel integral operator T_n are the same as the eigenvalues of the kernel matrix, and its eigenvectors are given by [Bengio et al., 2004]

$$\hat{\psi}_j^n = \frac{\sqrt{n}}{\hat{\lambda}_i^n} \sum_{j=1}^n u_{i,j} k(x_j, x) = \frac{\sqrt{n}}{\hat{\lambda}_i^n} u_i^T \kappa(x)$$

The values of $\hat{\psi}_j^n(x)$ at the points $x_1, x_2, ..., x_n$ equal the corresponding eigenvector of the kernel matrix K, $\hat{\psi}_j^n(x_i) = (u_i)_i$.

If we randomly sample $m < n$ data points $\{x_j\}_{j \in S}$ from the full dataset and then take the values of $\hat{\psi}_j^m,\ j = 1, 2, ..., m$ at all the points $x_1, x_2, ..., x_n$, and normalize by $\frac{1}{\sqrt{n}}$, we obtain the Nyström approximation [Williams and Seeger, 2001]

$$\widetilde{\lambda}_i = \frac{n}{m} \hat{\lambda}_i^m \tag{1}$$

$$\widetilde{u}_i = \sqrt{\frac{m}{n}} \frac{1}{\hat{\lambda}_i^m} K_{nm} u_i$$

Multiplying together the approximate eigenvectors and eigenvalues one so obtains an approximate kernel matrix $\widetilde{K} = K_{nm}K_{mm}^{-1}K_{mn}$ where K_{mm} contains the m subsampled rows and columns of K, K_{nm} contains the m subsampled columns, and K_{mn} is its transpose. The approximate kernel matrix can serve as a replacement of the original kernel matrix for improved computational efficiency for different kernel methods.

Kernel methods in machine learning look for functions in the reproducing kernel Hilbert space to be adapted to data

$$f(x) = \sum_{j=1}^{n} \alpha_j \langle \phi(x_j), \phi(x) \rangle_{\mathcal{H}} = \sum_{j=1}^{n} \alpha_i k(x_j, x)$$

where $\{\alpha_j\}_{j \in S}$ are parameters. The Nyström method may also be defined by restricting these functions to lie in the linear span of the m subsampled data points $\{\phi(x_i)\}_{i \in S}$, while using the full dataset of n points for estimation of the unknown parameters [Rudi et al., 2015]. For fixed S the linear span of $\{\phi(x_i)\}_{i \in S}$ is a closed subspace of $\mathcal{H}$ and so is a Hilbert space, which we will denote by $\mathcal{H}_S$ [Bollobás, 1999]. In other words, one looks for functions of the form

$$f(x) = \sum_{j \in S} \alpha_j \langle \phi(x_j), \phi(x) \rangle_{\mathcal{H}} = \sum_{j \in S} \alpha_i k(x_j, x)$$

where $\{\alpha_j\}_{j \in S}$ are parameters, that solve an empirical risk minimization problem based on all data points $\{x_i\}_{i=1}^{n}$.

After drawing the n observations $\{x_i\}_{i=1}^{n}$ independently from $\mathbb{P}_X$, the subset of m data points $\{x_i\}_{i \in S} = \{x_{i_1}, x_{i_2}, ..., x_{i_m}\}$ is randomly selected according to a specified distribution $p(S|\{x_i\}_{i=1}^{n})$. Before the data points are observed the elements in the subset are random variables $\{X_{i_1}, X_{i_2}, ..., X_{i_m}\}$. For notational convenience we will assume that the data points are reordered after the subsampling so that $\{x_i\}_{i \in S} = \{x_1, x_2, ..., x_m\}$.

Kernel PCA may be obtained by appealing to the $\ell^2(\mathbb{R})$ representation of a separable real Hilbert space and arranging the data points in $\mathcal{H}$ in a data matrix Φ with one data point occupying a row, which may then have an infinite number of columns. The principal components are then the eigenvectors of $\frac{1}{n}\Phi^T\Phi$ and the kernel matrix can be written as $K = \Phi\Phi^T$. The mean can be subtracted in the RKHS (the feature space) through [Schölkopf et al., 1998]

$$K' = (\Phi - \mathbb{1}_n\Phi)(\Phi - \mathbb{1}_n\Phi)^T = K - \mathbb{1}_nK - K\mathbb{1}_n + \mathbb{1}_nK\mathbb{1}_n$$

where $\mathbb{1}_n$ is a matrix for which $(\mathbb{1}_n)_{i,j} = \frac{1}{n}$. The eigenvalues of $K' = Q\Lambda Q^T$ scaled by $\frac{1}{n}$ then measure the variance of the data projected onto each individual principal component. Its eigenvectors Q are proportional to the principal scores – the principal scores are given by $S = Q\Lambda^{1/2}$. By the singular value decomposition $\Phi - \mathbb{1}_n\Phi = Q\Sigma E^T$, where $\Lambda = \Sigma^2$, the principal scores of a *new* data point x^* which is centred in feature space is given by

$$\begin{aligned} w^* &= ((\phi(x^*) - \mathbf{1}_n\Phi)E)^T = ((\phi(x^*) - \mathbf{1}_n\Phi)(\Phi - \mathbb{1}_n\Phi)^T Q\Lambda^{-1/2})^T \\ &= ((\kappa(x^*)^T - \kappa(x^*)^T\mathbb{1}_n - \mathbf{1}_nK + \mathbf{1}_nK\mathbb{1}_n)Q\Lambda^{-1/2})^T \\ &= \Lambda^{-1/2}Q^T(\kappa(x^*) - \mathbb{1}_n\kappa(x^*) - K\mathbf{1}_n + \mathbb{1}_nK\mathbf{1}_n) =: \Lambda^{-1/2}Q^T\kappa'(x^*) = S^{-1}\kappa'(x^*) \end{aligned}$$

where $\phi(x_i)$ is an element in $\ell^2(\mathbb{R})$ as a row vector, $\mathbf{1}_n$ is a length-n column vector with each element equal to $\frac{1}{n}$ and $\kappa(x) = (k(x_1, x), k(x_2, x), ..., k(x_n, x))^T$.

Using this formula to calculate the scores for the *original* data points we get that $\kappa'(x^*)$ becomes K' and obtain $w^{*T} = K'Q\Lambda^{-1/2} = Q\Lambda Q^T Q\Lambda^{-1/2} = Q\Lambda^{1/2}$ and so as expected we recover the previous expression for the principal scores.

When applying PCA to a real-world problem it is often appropriate to normalize the input variables to have variance 1, so as to make the analysis independent of arbitrary changes of units in the data. Otherwise the variables with higher variance will dominate the principal components and comparisons between variables become difficult. This normalization will often also be appropriate for kernel PCA and we do this for the experimental analysis (Section 7). The centring of variables in the feature space does not guarantee that the input variables become centred.

Multi-dimensional scaling (MDS) finds a lower-dimensional representation of data from a matrix of distances between data points [Hout et al., 2013]. MDS is equivalent to kernel PCA when the kernel is *isotropic*, i.e. on the form $f(\|x - y\|)$ for some function f [Williams, 2002]. Therefore, theoretical or practical results for kernel PCA are often also applicable to MDS.

The approximate eigenvalues and eigenvectors from the Nyström method in Equation (1) applied to the centred kernel matrix may be used to define an approximate kernel PCA. However, these approximate principal scores are not orthogonal (i.e. uncorrelated), so they do not define true PCA, and the eigenvalues do not describe the variance captured by the principal components. There is a need for another way to derive kernel PCA with the Nyström method.

4. Kernel PCA with the Nyström method

In this section we present kernel PCA with the Nyström method, which provides an efficient and flexible technique for non-linear PCA. We present the corresponding quantities that are defined for linear PCA and are useful for data exploration and application of the method in downstream tasks

(1) a set of orthogonal principal components with unit length in the linear span of the subsampled data points in $\mathcal{H}$ (denoted $\mathcal{H}_S$),

(2) the variance of the data along each of these directions, termed the explained variance,

(3) the reconstruction error of the data onto the principal components,

(4) a set of uncorrelated principal scores with the weightings of the data points on the principal components, and,

(5) the principal scores of a new data point with respect to the existing principal components

For standard kernel PCA (2) and (3) are the same, but with the Nyström method they are different, since the principal components will not span the entire data.

We first present the principal components, explained variance and scores for a dataset in the following theorem

Theorem 1 (Nyström kernel PCA)**.** *Let $(\widetilde{\lambda}_j, v_j)$ be the eigenpairs and $V\widetilde{\Lambda}V^T$ be the eigendecomposition of*

$$\frac{1}{n}\widetilde{K}' = \frac{1}{n}K'^{-1/2}_{mm}K'_{mn}K'_{nm}K'^{-1/2}_{mm} \tag{2}$$

where

$$K'_{mn} = K_{mn} - K_{mn}\mathbb{1}_n - \mathbb{1}_n^{m,n}\widetilde{K} + \mathbb{1}_n^{m,n}\widetilde{K}\mathbb{1}_n$$

$$K'_{mm} = K_{mm} - \mathbb{1}_n^{m,n}K_{nm} - K_{mn}\mathbb{1}_n^{n,m} + \mathbb{1}_n^{m,n}\widetilde{K}\mathbb{1}_n^{m,n}$$

with $\widetilde{K} = K_{nm}K_{mm}^{-1}K_{mn}$ and where $\mathbb{1}_n$, $\mathbb{1}_n^{n,m}$ and $\mathbb{1}_n^{m,n}$ are $n \times n$, $n \times m$ and $m \times n$ matrices respectively with each element equal to $\frac{1}{n}$.

The perpendicular intersecting lines $\phi_0 + \langle\widetilde{\phi}_j\rangle$, $j = 1, 2, ..., m$ in $\mathcal{H}_S$ along which the variance of the data is successively maximized, where the orthonormal vectors $\{\widetilde{\phi}_j\}_{j=1}^m$ are termed the principal components, are given by

$$\begin{aligned} \phi_0 &= \frac{1}{n}K_{nm}K_{mm}^{-1}\kappa_m(x) \\ \widetilde{\phi}_j &= \sum_{k=1}^{m} u_{j,k}\left(k(x_k, x) - \phi_0\right) \end{aligned} \tag{3}$$

and the variances along these directions are $\{\widetilde{\lambda}_j\}_{j=1}^m$, termed the principal values or explained variance, where $\kappa_m(x) = (k(x_1, x), k(x_2, x), ..., k(x_m, x))^T$, $u_j = K'^{-1/2}_{mm}v_j$ and $U = K'^{-1/2}_{mm}V$.

The projection coefficients of the centred data points onto the principal components, termed the principal scores, are given by

$$W = K'_{nm}U = K'_{nm}K'^{-1/2}_{mm}V$$

where each row of W contains the scores of one data point onto the principal components.

The principal scores of a new data point x^* is given by

$$w^* = U^T(\kappa_m(x^*) - K_{mn}\mathbf{1}_n - \mathbb{1}_n^{m,n}K_{nm}K_{mm}^{-1}\kappa_m(x^*) + \mathbb{1}_n^{m,n}\widetilde{K}\mathbf{1}_n) = U^T\,\widetilde{\kappa}(x^*)$$

where $\mathbf{1}_n$ is a length-n column vector given by $\mathbf{1}_n = (\frac{1}{n}, \frac{1}{n}, \ ..., \ \frac{1}{n})^T$.

The principal components can be seen as defining new variables through linear combinations of the existing variables that have successively maximized variance and that are uncorrelated. The values of these new variables are given by the principal scores, which represent the data in a new coordinate system defined by the principal components as a new basis for the space. As such, the principal scores can be used as a drop-in replacement for the original data in arbitrary supervised or unsupervised learning methods, including after removing the scores corresponding to principal components with smaller eigenvalues. Please see Section 8 for an example of this.

To see that these new variables are uncorrelated also with the Nyström method we note that

$$W^T W = V^T K'^{-1/2}_{mm} K'_{mn} K'_{nm} K'^{-1/2}_{mm} V = n V^T V \widetilde{\Lambda} V^T V = n\widetilde{\Lambda}$$

which is a diagonal matrix.

The computational complexity of the method is $\mathcal{O}(nm^2)$ in time, which is the same as the Nyström method applied to regression. Centring of the matrix K_{nm} can be accomplished in $\mathcal{O}(m^3 + nm)$ operations, and so the centring in the proposed method adds no additional time requirements to the dominant $\mathcal{O}(nm^2)$ factor. We refer to the software implementation for full details.

The Nyström method approximates the corresponding full method, so when $m = n$ we should recover standard kernel PCA. In this case $\widetilde{K} = KK^{-1}K = K$ and as expected $K'_{mm} = K'_{nm} = K'$ and

$$K'^{-1/2}_{mm} K'_{mn} K'_{nm} K'^{-1/2}_{mm} = K'$$

and the scores are equal to $W = K'^{1/2} V = \sqrt{n}\, V \widetilde{\Lambda}^{1/2} V^T V = \sqrt{n}\, V \widetilde{\Lambda}^{1/2} = Q\Lambda^{1/2}$, which we know to be the scores for standard kernel PCA.

The scores of new data points are important when measuring the accuracy of PCA with a test set of hold-out data points, for example using the reconstruction error (Section 7), or when applying PCA as a preprocessing step for supervised learning methods and one wishes to create predictions for new data points, such as in principal component regression (Section 8).

If the data points are assumed to have zero mean in feature space then the matrices K'_{nm} and K'_{mm} may be replaced by K_{nm} and K_{mm} and the vector $\widetilde{\kappa}(x)$ by $\kappa_m(x)$. The principal components are then given by $\widetilde{\phi}_j = \sum_{k=1}^{m} u_{j,k} k(x_k, x)$.

The smallest $m - d$ Nyström eigenvalues $\sum_{j=d+1}^{m} \widetilde{\lambda}_j$ measure the residual variance of the data points *within* $\mathcal{H}_S$ and correspond to the reconstruction error $\frac{1}{n}\sum_{i=1}^{n} \|P_{\widetilde{V}_d} z'_i - P_{\mathcal{H}_S} z'_i\|^2_{\mathcal{H}}$, where $\widetilde{V}_d = \operatorname{span}\{\{\widetilde{\phi}_k\}_{k=1}^{d}\}$. The full reconstruction error with respect to the top d Nyström principal components is given by

$$R_n(\widetilde{V}_d) = \frac{1}{n}\sum_{i=1}^{n} \|z'_i - P_{\widetilde{V}_d} z'_i\|^2_{\mathcal{H}} = \frac{1}{n}\operatorname{Tr}(K') - \sum_{j=1}^{d} \widetilde{\lambda}_j \tag{4}$$

where $\operatorname{Tr}(\cdot)$ is the trace and $\frac{1}{n}\operatorname{Tr}(K')$ is the variance of the full dataset in $\mathcal{H}$. From Theorem 1 we know that this is the smallest reconstruction error among all d-dimensional subspaces in $\mathcal{H}_S$.

Calculation of this quantity is $\mathcal{O}(n^2)$ due to the centring of K. However, it can be approximated for example by subtracting the mean of K_{nm} instead of the mean of K, which becomes $\mathcal{O}(nm)$. This is included as an option in the software package accompanying the paper[3]. Please see Section 7 for further details.

[3] https://github.com/fredhallgren/nystrompca

Note that the reconstruction error above in Equation (4) is slightly different from the reconstruction error of the uncentred data points with respect to the affine subspace $\phi_0 + \widetilde{V}_d$, which becomes $\frac{1}{n}\sum_{i=1}^{n} \|(z_i - \phi_0) - P_{\widetilde{V}_d}(z_i - \phi_0)\|_{\mathcal{H}}^2 = \frac{1}{n}\sum_{i=1}^{n} \|(z_i - \phi_0) - P_{\widetilde{V}_d} z_i'\|_{\mathcal{H}}^2$. Both reconstruction errors are at a minimum for the proposed method.

Another quantity of interest is the reconstruction error of the full dataset on the eigenspace of the subset of m data points. Creating PCA from a random subset of m data points to describe the full dataset will be termed *Subset PCA*. We may use the same centring as for the Nyström method and maintain the $\mathcal{O}(m^3)$ time complexity – that is to say we use the mean of the n data points projected onto $\mathcal{H}_S$. This also ensures that the amount of variance captured is the same whether we project the centred data onto the principal components, or the uncentred data onto the lines translated from the origin. The principal components will then be given by, for $j = 1, 2, ..., m$

$$\hat{\phi}_j^{m,n} = \sum_{k=1}^{m} u_{j,k}^{m}(k(x_k, x) - \phi_0)$$

where u_j^m is the jth eigenvector of $\frac{1}{m}K'_{mm}$. The variance of the full data captured by these principal components and the associated reconstruction error are presented in the following theorem

Theorem 2 (Subset PCA). *The variance of the dataset $\{\phi(x_i)\}_{i=1}^{n}$ along the jth principal component $\hat{\phi}_j^{m,n}$ is given by*

$$\hat{\lambda}_j^{m,n} = \frac{1}{n}\sum_{i=1}^{n} \|P_{\hat{\phi}_j^{m,n}} z_i'\|_{\mathcal{H}}^2 = \frac{1}{n \cdot m\hat{\lambda}_j^m} u_j^{m\,T} K'_{mn} K'_{nm} u_j^m$$

where $(\hat{\lambda}_j^m, u_j^m)$ is the jth eigenpair of $\frac{1}{m}K'_{mm}$.

The reconstruction error of the full dataset onto the corresponding d-dimensional PCA subspace is

$$R_n(\hat{V}_d^m) = \frac{1}{n}\sum_{i=1}^{n} \|z_i' - P_{\hat{V}_d^m} z_i'\|_{\mathcal{H}}^2 = \frac{1}{n}\mathrm{Tr}(K') - \frac{1}{n \cdot m}\mathrm{Tr}(K'_{nm} U_d^m \Lambda_d^{m\,-1} U_d^{m\,T} K'_{mn})$$

where $U_d^m \Lambda_d^m U_d^{m\,T}$ is the truncated eigendecomposition of $\frac{1}{m}K'_{mm}$.

As expected, if $n = m = d$ then the reconstruction error is zero.

The method proposed in this section for efficient kernel PCA can also be applied to improve the scalability of MDS when these two methods are equivalent, as outlined in Section 3.

5. Prelude: A special case

Before studying the statistical accuracy of kernel PCA with Nyström method through a confidence bound we present a majorization relation between Nyström and Subset PCA and consider the special case when the PCA dimension equals the number of subsampled data points, $d = m$. In this case the reconstruction error for the Nyström method is the same as Subset PCA, both for the empirical and true reconstruction errors.

Proposition 1. *We have the following majorization relation for the empirical error*

$$(\widetilde{\lambda}_1, \widetilde{\lambda}_2, ..., \widetilde{\lambda}_m) \succ (\hat{\lambda}_1^{m,n}, \hat{\lambda}_2^{m,n}, ..., \hat{\lambda}_m^{m,n})$$

The majorization is strict in the sense that $\widetilde{\lambda}_{<d} > \hat{\lambda}_{<d}^{m,n}$ for $d < m$, by the assumption of a continuous data distribution.

A direct consequence of the proposition is that

$$R_n(\widetilde{V}_m) = R_n(\hat{V}_m^m)$$

For the true reconstruction error we consider the case where the sampling of the Nyström subset occurs independently of the values of the data points

Proposition 2. *Let $d = m$ and let the Nyström subset be sampled according to $p(S \,|\, x_1, x_2, ..., x_n)$. Then if*

$$p(S \,|\, x_1, x_2, ..., x_n) = p(S)$$

i.e. the subsampling is independent of the data, we have

$$R(\widetilde{V}_m) = R(\hat{V}_m^m)$$

The above proposition includes the common case of uniform sampling for the Nyström subset. It holds whether the n data points are considered fixed or unobserved.

From the above propositions we can conclude that if retaining all the Nyström principal components then there is no gain in accuracy compared to Subset PCA from the perspective of the reconstruction error. However, for a smaller PCA dimension the Nyström method will perform strictly better than PCA directly from the subset. Furthermore, other strategies for sampling of the subset may lead to a higher accuracy for the Nyström method even when $d = m$.

6. Statistical accuracy of Nyström kernel PCA

In this section we provide a high probability confidence bound on the empirical reconstruction error of kernel PCA with the Nyström method versus the one for full kernel PCA. In line with most results on the statistical accuracy on kernel PCA we assume that data has zero mean in feature space.

The actual difference between the reconstruction errors of the Nyström method and standard kernel PCA for a specific dataset is given by

$$R_n(\widetilde{V}_d) - R_n(\hat{V}_d) = \frac{1}{n}\mathrm{Tr}(K) - \sum_{j=1}^{d} \widetilde{\lambda}_j - \sum_{j=d+1}^{m} \hat{\lambda}_j^n \ = \ \hat{\lambda}_{<d}^n \ - \ \widetilde{\lambda}_{<d}$$

However, the eigenvalues $\hat{\lambda}_j^n$ of $\frac{1}{n}K$ are not available – if they were there would be no need to apply the Nyström method. When the Nyström method is being considered for a problem then the size of the data n is very large and calculating the full kernel matrix K, let alone its eigendecomposition, is prohibitively expensive.

At a minimum, any measure of accuracy should not be more computationally demanding than the method itself, which is $\mathcal{O}(nm^2)$. We present a bound that does not require that we have observed the entire dataset, only the subset $x_1, x_2, ..., x_m$. It takes $\mathcal{O}(m^3)$ time to calculate and is $\mathcal{O}(m^2)$ in memory. It holds for any subsampling distribution.

Theorem 3 (Confidence bound). *With confidence $1 - 2e^{-\delta}$ for $d = 1, 2, ..., m-1$ and $\{x_i\}_{i \notin S} \sim p_X(x)$, where $B := \sup_x k(x,x)$, $\Phi(\cdot)$ is the standard normal cumulative distribution function, $\{\hat{\lambda}_j^m\}_{j=1}^m$ are the eigenvalues of the kernel matrix $\frac{1}{m}K_{mm}$ from the Nyström subset, and*

$$D \ := \frac{n-m}{n}\left(\frac{B\sqrt{2\delta}}{\sqrt{n-m}} + \frac{B^2}{\sqrt{m}}\left(\sqrt{2\log 2} + 2\sqrt{2\pi}\,\Phi\left(-\sqrt{2\log 2}\right)\right)\right)$$

$$D_k := \frac{D^2}{\left(\hat{\lambda}_k^m - \hat{\lambda}_{k+1}^m\right)^2} \wedge 1$$

we have

$$R_n(\widetilde{V}_d) - R_n(\hat{V}_d) \leq \sum_{j=1}^{d} \hat{\lambda}_j^m \cdot D_j + D \cdot \max_{1\leq k\leq d} D_k$$

The bound does not require that we have observed the entire sample. For example, if data is generated sequentially and iid from $p_X(x)$ then picking the first m points for the Nyström subset is equivalent to sampling all points and then selecting m points uniformly (in the sense that the data points in the subset have the same distribution in both instances).

If data is stored on disk, and reading from disk is expensive, then only m records need to be read in order to calculate the bound, assuming this can be done in such a way as to respect the sampling distribution of the subset of data points. In many implementations of the SQL query language, including MySQL and PostgreSQL, this would correspond to appending `LIMIT(m)` to the end of the query, which interrupts it after finding the first m records [Beaulieu, 2020]. This may particularly improve performance if the query itself is time-consuming.

The bound becomes infinite if $k(x,x)$ is not bounded for all x. However, every kernel can be made bounded for example through the transformation

$$k'(x,y) := \frac{k(x,y)}{\sqrt{k(x,x)k(y,y)}} \tag{5}$$

which has $\sup_x k(x,x) = 1$.

Proof outline. A proof outline is as follows. Please see the appendix for a full proof.

1. Rewrite the difference in reconstruction errors in terms of the eigenpairs of the empirical operators C_n and C_m, to obtain
$$R_n(\widetilde{V}_d) - R_n(\hat{V}_d) \leq \sum_{j=1}^{d} \hat{\lambda}_j^n \left(1 - \langle \hat{\phi}_j^n, \hat{\phi}_j^m \rangle_{\mathcal{H}}^2\right)$$
2. Apply the Davis-Kahan theorem to convert the angle between the eigenvectors into a difference between successive eigenvalues of C_m and the norm of the difference between the empirical operators $\|C_n - C_m\|$
3. Convert the unknown eigenvalues $\hat{\lambda}_j^n$ into the ones based on the observed data $\hat{\lambda}_j^m$ plus the difference $\|C_n - C_m\|$, using Lidskii's inequality
4. Now $\|C_n - C_m\|$ is the only random and unknown quantity left. Split it up into two independent terms through
$$\|C_n - C_m\| \leq \frac{n-m}{n}(\|C_{n-m} - \mathbb{E}[C_{n-m}]\| + \|C_m - \mathbb{E}[C_m]\|)$$
where $C_{n-m} = \frac{1}{n-m}\sum_{i=m+1}^{n} z_i \otimes z_i$
5. Apply Hoeffding's inequality in Banach spaces to the first term, obtaining
$$\mathbb{P}\left(\|C_{n-m} - \mathbb{E}[C_{n-m}]\| \leq \sqrt{2\delta}B/\sqrt{n-m}\right) \geq 1 - 2e^{-\delta}$$
6. Write the second term in terms of the evaluation operator, then apply Hoeffding's inequality to the random part, and then calculate its expectation based on the obtained distribution function

6.1. A corollary.

From the proof of Theorem 3 one can deduce sharper versions of Theorem 7 and Propositions 10 and 11 from Rosasco et al. [2010], by a factor 2 or 4, although at the expense of slightly longer proofs. These follow from showing that since the covariance operator C and its empirical equivalent C_n are positive, then $\|C - C_n\|_{\mathrm{HS}(\mathcal{H})}$ is bounded by $\sup_x k(x,x)$, rather than $2\sup_x k(x,x)$.

For Theorem 7, the sharper result states that with probability at least $1 - 2e^{-\delta}$ we have

$$\|C - C_n\|_{\mathrm{HS}(\mathcal{H})} \leq \frac{B\sqrt{2\delta}}{\sqrt{n}}$$

The sharper version of Proposition 10 states that with probability $1 - 2e^{-\delta}$

$$\sum_{j=1}^{\infty} \left(\lambda_j - \hat{\lambda}_j^n\right)^2 \leq \frac{2B^2\delta}{n}$$

$$\sup_j |\lambda_j - \hat{\lambda}_j^n| \leq \frac{B\sqrt{2\delta}}{\sqrt{n}}$$

And for Proposition 11 we obtain that also with probability $1 - 2e^{-\delta}$

$$\left|\sum_{j=1}^{\infty} \lambda_j - \sum_{j=1}^{n} \hat{\lambda}_j^n\right| = |\mathrm{Tr}(C) - \mathrm{Tr}(C_n)| \leq \frac{B\sqrt{2\delta}}{\sqrt{n}}$$

7. Experimental analysis

In this section we illustrate the method and bound through experiments on real-world datasets with different kernel functions. We first compare the proposed method to a number of other unsupervised learning methods by measuring the reconstruction error on hold-out data sets. We then evaluate the bound and compare it to the actual errors and the errors for Subset PCA.

The methods and experiments are implemented in the Python programming language and the source code is available at `https://github.com/fredhallgren/nystrompca`. The package can be installed with one simple command using the Python package manager[4]. It includes a command-line tool to run the different experiments with different parameter values and kernel functions.

For purposes of reproducibility the computer experiments allow for setting the random seed of the random number generator [Robert and Casella, 2004], to produce exactly the same results every time the experiments are run. Other than the random sampling of the Nyström subset, randomness is also present in the splitting of data into training and test sets.

The principal components are unique only up to a sign, so in the package we switch the sign of the scores and components such that the range of values in each dimension of the scores is mostly

[4] `pip install nystrompca`

positive. This will ensure that we will get exactly the same values for the scores and components every time we run the algorithm.

We use different datasets from the UCI Machine Learning Repository [Dua and Graff, 2017]. Dimensionality reduction can be particularly important for high-dimensional data, so we include a number of such datasets. We use the simulated magic gamma telescope dataset, the yeast dataset, containing cellular protein location sites for fungi, the cardicotocography dataset, with heart measurements, the segmentation dataset containing various data on images, the drug dataset with personality traits and drug consumption, the digits dataset with flattened 8×8 pixel grayscale images, and two bag-of-words datasets with bag-of-words vectors of articles from www.dailykos.com and NIPS papers, respectively. We tabulate some information on the datasets used in one or both of the experiments below in Table 1, where the number of features is before any data transformation. For comparability we cut each dataset to 1000 data points when running the experiments. In both experiments we use a Nyström subset of size $m = 100$ which we sample uniformly without replacement and we use the same sampled subset for both Nyström PCA and Subset PCA.

	Dataset	*Data size*	*Number of attributes*
1	magic	19020	11
2	yeast	1484	8
3	cardiotocography	2126	23
4	segmentation	2310	19
5	drug	1885	32
6	digits	5620	64
7	dailykos	3430	6906
8	nips	1500	12419

TABLE 1. Datasets used

We convert ordinal variables to integers and categorical variables to discrete ones through one-hot encoding. We treat discrete numerical variables in the data as continuous for the purposes of PCA. We remove any date or time variables. We also remove variables that are constant. These will differ depending on how many data points we include in the total dataset when we run the experiments.

We normalize the input data to have mean zero and variance one. Note that this does not mean that data has zero mean in the feature space. As previously mentioned, normalizing the input data makes the analysis independent of the units used to measure the variables and unaffected by the scale of the variables, which may otherwise dominate the PCA results. Furthermore, it makes it easier to compare results across different data sets and kernel functions and makes the same kernel parameters appropriate for all data sets.

We cut eigenvalues that are smaller than 10^{-12} when performing matrix inversions to improve the condition number of the matrix. We also remove any negative eigenvalues – in theory all kernel matrices will be positive definitive, however numerical inaccuracies may occasionally lead to small negative eigenvalues in practice.

We use three different kernel functions, the radial basis functions (RBF), polynomial and Cauchy kernels, summarized below in Table 2. The software package includes a number of additional kernel functions that can be used when running either of the experiments.

Kernel	*Functional form* $k(x, y)$	*Parameters*	*Bound* $\sup_x k(x, x)$
RBF	$\exp\left\{-\frac{\|x-y\|^2}{\sigma^2}\right\}$	$\sigma \in \mathbb{R}_+ \setminus \{0\}$	1
Polynomial	$(\langle x, y\rangle + R)^d$	$R \in \mathbb{R},\ d \in \mathbb{N}$	∞
Cauchy	$\frac{1}{1+\|x-y\|^2/\sigma^2}$	$\sigma \in \mathbb{R}_+ \setminus \{0\}$	1

TABLE 2. Kernel functions used

7.1. **Methods comparison.** We compare the proposed method to other unsupervised learning techniques to evaluate its behaviour. We compare with linear PCA, full kernel PCA, sparse PCA [Wang et al., 2016], locally linear embeddings (LLE), a manifold method [Roweis and Saul, 2000] and independent component analysis (ICA) [Hyvärinen and Oja, 2000]. We run the methods for all the datasets in Table 1 above. We split each dataset randomly in half, fitting the methods on one half and then evaluating them on the other half. We compare the fraction of variances captured for the different methods for different dimensions. For kernel PCA and Nyström kernel PCA we measure the variances captured in the RKHS and not in the input space. For this experiment we only display the results for the RBF kernel. We calculate the bandwidth parameter as the mean distance between pairs of data points, which is a common heuristic for the RBF kernel. Using all pairs of data points is quadratic in the total number of data points, so we only use the data points in the Nyström subset. Please see Table 3 for the full results, where we have set the random seed to 1. Sparse PCA is NP-hard with respect to the data dimension and so is not computationally feasible for very high-dimensional data. Therefore we don't run it for all the datasets. To run these experiments using the supplied command-line tool one would do

```
> nystrompca methods --seed 1
```

Note that the purpose of each of these methods is not necessarily to capture as much variance as possible, however it can still be enlightening to contrast this quantity between different methods. Furthermore, since linear PCA acts in the input space and kernel PCA and its derivations act in the feature space, comparison of the amount of variance captured are not necessarily clear-cut.

Nyström kernel PCA generally captures more variance than Subset PCA, apart from the two bag-of-words datasets (`dailykos` and `nips`). Since we are calculating the reconstruction error on a hold-out dataset it's possible that Subset PCA achieves better performance – we know this to be impossible for the training dataset by Proposition 1. For datasets with a small number of dimensions standard linear PCA captures the most amount of variance whilst being simpler and more computationally efficient. For all datasets the performance of Nyström kernel PCA is very close to the method it is attempting to approximate, despite being many times more efficient.

Dataset	d	*Subset PCA*	*Nyström*	*Kernel PCA*	*Linear PCA*	*Sparse PCA*	*LLE*	*ICA*
magic								
	1	0.2116	0.2268	0.2274	0.5126	0.5056	0.1050	0.0937
	2	0.3459	0.3593	0.3610	0.6257	0.6090	0.2223	0.1798
	3	0.4089	0.4246	0.4269	0.7155	0.7080	0.3035	0.2630
	4	0.4752	0.4912	0.4923	0.7929	0.7342	0.3805	0.3488
	5	0.5292	0.5506	0.5539	0.8635	0.7841	0.5017	0.3488
	6	0.5584	0.5875	0.5933	0.9250	0.8407	0.5966	0.5535
	7	0.5897	0.6230	0.6291	0.9633	0.8647	0.7055	0.6962
	8	0.6126	0.6467	0.6540	0.9816	0.9008	0.7884	0.6962
	9	0.6300	0.6699	0.6781	0.9978	0.9603	0.9048	0.6962
	10	0.6459	0.6891	0.6982	1.0000	0.9803	1.0000	1.0000
yeast								
	1	0.1264	0.1387	0.1396	0.1214	0.1181	0.0339	0.0431
	2	0.2336	0.2600	0.2614	0.2177	0.2080	0.0784	0.0911
	3	0.3197	0.3756	0.3777	0.2773	0.2648	0.1323	0.1223
	4	0.4391	0.4521	0.4550	0.4057	0.4051	0.1848	0.1977
	5	0.4804	0.4998	0.5037	0.5052	0.4958	0.2299	0.2598
	6	0.5238	0.5435	0.5474	0.5869	0.5972	0.3271	0.3149
	7	0.5559	0.5771	0.5839	0.6484	0.6546	0.3867	0.3655
	8	0.5718	0.6102	0.6169	0.7057	0.7098	0.4344	0.4145
	9	0.6022	0.6439	0.6509	0.7520	0.7505	0.4720	0.4569
	10	0.6346	0.6736	0.6828	0.7988	0.7972	0.5252	0.5056
cardiotocography								
	1	0.1329	0.1351	0.1357	0.2223	-	0.0509	0.0260
	2	0.2183	0.2284	0.2306	0.3564	-	0.0795	0.0521
	3	0.2833	0.3012	0.3043	0.4577	-	0.0928	0.0765
	4	0.3374	0.3556	0.3594	0.5258	-	0.1241	0.1019
	5	0.3766	0.3983	0.4029	0.5782	-	0.1449	0.1264
	6	0.4043	0.4346	0.4399	0.6259	-	0.1763	0.1539
	7	0.4342	0.4672	0.4738	0.6636	-	0.2149	0.1790
	8	0.4594	0.5001	0.5061	0.7013	-	0.2468	0.2051
	9	0.4791	0.5217	0.5312	0.7342	-	0.2682	0.2296
	10	0.5056	0.5467	0.5571	0.7687	-	0.3122	0.2576
segmentation								
	1	0.2563	0.2620	0.2621	0.3107	0.3044	0.0222	0.0387
	2	0.3871	0.3952	0.3955	0.5541	0.5468	0.0580	0.1223
	3	0.4988	0.5040	0.5044	0.6369	0.6165	0.1555	0.1573
	4	0.5494	0.5556	0.5565	0.6787	0.6539	0.1885	0.1934
	5	0.6017	0.6039	0.6048	0.7274	0.6971	0.2690	0.2429
	6	0.6434	0.6535	0.6543	0.7930	0.7649	0.3116	0.3109
	7	0.6785	0.6886	0.6921	0.8427	0.7969	0.3733	0.3676
	8	0.6967	0.7102	0.7139	0.8816	0.8190	0.4278	0.4284
	9	0.7147	0.7295	0.7332	0.9087	0.8558	0.4412	0.4739
	10	0.7314	0.7469	0.7513	0.9665	0.9127	0.5149	0.6188

Table continues on the next page

Dataset	*d*	*Subset PCA*	*Nyström*	*Kernel PCA*	*Linear PCA*	*Sparse PCA*	*LLE*	*ICA*
drug								
	1	0.1342	0.1398	0.1425	0.2316	-	0.0376	0.0278
	2	0.1688	0.1791	0.1837	0.3031	-	0.0602	0.0573
	3	0.2010	0.2219	0.2284	0.3594	-	0.1104	0.0874
	4	0.2261	0.2464	0.2538	0.4059	-	0.1226	0.1149
	5	0.2446	0.2734	0.2827	0.4462	-	0.1559	0.1418
	6	0.2773	0.3006	0.3105	0.4846	-	0.1944	0.1699
	7	0.2970	0.3217	0.3338	0.5094	-	0.2362	0.1905
	8	0.3162	0.3487	0.3631	0.5515	-	0.3044	0.2276
	9	0.3330	0.3648	0.3805	0.5791	-	0.3405	0.2530
	10	0.3483	0.3807	0.3977	0.6045	-	0.3771	0.2766
digits								
	1	0.0715	0.0734	0.0749	0.1261	-	0.0190	0.0148
	2	0.1459	0.1542	0.1572	0.2261	-	0.0310	0.0289
	3	0.2072	0.2163	0.2210	0.3156	-	0.0516	0.0442
	4	0.2530	0.2684	0.2754	0.3914	-	0.0730	0.0595
	5	0.2960	0.3093	0.3183	0.4526	-	0.0876	0.0772
	6	0.3220	0.3403	0.3509	0.4946	-	0.1208	0.0914
	7	0.3502	0.3718	0.3847	0.5353	-	0.1479	0.1067
	8	0.3730	0.3976	0.4122	0.5693	-	0.1590	0.1221
	9	0.3984	0.4203	0.4371	0.6018	-	0.2006	0.1373
	10	0.4113	0.4425	0.4624	0.6321	-	0.2258	0.1534
dailykos								
	1	0.0879	0.0856	0.0843	0.0079	-	0.0086	0.0033
	2	0.0917	0.0915	0.0914	0.0096	-	0.0094	0.0040
	3	0.0918	0.0915	0.0926	0.0109	-	0.0147	0.0048
	4	0.0918	0.0915	0.0932	0.0119	-	0.0155	0.0054
	5	0.0918	0.0915	0.0935	0.0126	-	0.0174	0.0058
	6	0.0918	0.0915	0.0939	0.0131	-	0.0201	0.0062
	7	0.0918	0.0915	0.0939	0.0135	-	0.0218	0.0065
	8	0.0918	0.0915	0.0940	0.0140	-	0.0262	0.0069
	9	0.0919	0.0916	0.0940	0.0143	-	0.0335	0.0071
	10	0.0921	0.0916	0.0940	0.0147	-	0.0368	0.0074
nips								
	1	0.1035	0.0479	0.0435	0.0011	-	0.0039	0.0046
	2	0.1036	0.0480	0.0439	0.0024	-	0.0084	0.0114
	3	0.1037	0.0482	0.0443	0.0034	-	0.0088	0.0164
	4	0.1037	0.0482	0.0445	0.0037	-	0.0088	0.0187
	5	0.1038	0.0482	0.0446	0.0039	-	0.0089	0.0195
	6	0.1040	0.0483	0.0447	0.0042	-	0.0133	0.0216
	7	0.1040	0.0483	0.0448	0.0045	-	0.0161	0.0242
	8	0.1040	0.0483	0.0449	0.0049	-	0.0189	0.0267
	9	0.1041	0.0484	0.0451	0.0051	-	0.0217	0.0281
	10	0.1041	0.0484	0.0451	0.0054	-	0.0236	0.0305

TABLE 3. Comparison of the variance captured by different dimensionality reduction methods across the maximum dimension d

Calculation of Nyström kernel PCA takes on average 0.988 seconds across the eight datasets on an AWS EC2 m5.large instance with an Intel Xeon® Platinum 8175M CPU[5] running Ubuntu Server 20.04 with Linux kernel version 5.4, versus 2.753 seconds for full kernel PCA ($n = 500$, $m = 100$). In both instances the kernel matrices are created in Python whilst the eigendecomposition uses built-in LAPACK routines written in Fortran[6]. For these values of n and m the cubic time complexity is not attained and the constant, linear and quadratic factors are still important.

7.2. **Bound evaluation.** To demonstrate and evaluate the confidence bound as applied to data we compare it to the actual difference between the Nyström reconstruction error and the standard one, as well to the difference between the standard reconstruction error and the reconstruction error for Subset PCA. These quantities are generally not available when applying the Nyström method since they depend on the eigenvalues of the full kernel matrix, but we calculate them here for purposes of illustration.

We calculate the bound for PCA dimensions 1 through 10 and use a confidence level of 0.9 when calculating the bound. We run the experiments for multiple samples of the Nyström subset and plot the averages for the relevant quantities using 100 samples. The individual runs for different samples are run in parallel to leverage multi-core CPUs.

We plot the results of the experiments for the first four datasets in Table 1 and the kernels in Table 2 for different PCA dimensions below in Figures 1, 2 and 3. For the RBF and Cauchy kernels we set the bandwidth to $\sigma = 1$ and for the polynomial kernel we use $R = 1$ and $d = 2$. The RBF and Cauchy kernels are bounded by $\sup_x k(x, x) = 1$ and we normalize the polynomial kernel according to Equation (5) before applying it in the experiments. Each plot contains

(1) The values of the confidence bound (“Conf. bound”)

(2) The difference between the Nyström PCA and standard errors $R_n(\widetilde{V}_d) - R_n(\hat{V}_d^n)$ (“Nyström diff.”)

(3) The difference between the Subset PCA and standard errors $R_n(\hat{V}_d^m) - R_n(\hat{V}_d^n)$ (“Subset diff.”)

Running the bound evaluation experiments with the command-line tool can be accomplished with the following command

```
> nystrompca bound
```

Both the Nyström difference, the subset difference and the bound increase as the PCA dimension increases. The bound increases more rapidly as the PCA dimension increases from low values, but levels out for larger values as the tail eigenvalues decrease.

[5] https://aws.amazon.com/ec2/instance-types/

[6] https://numpy.org/devdocs/reference/generated/numpy.linalg.eigh.html

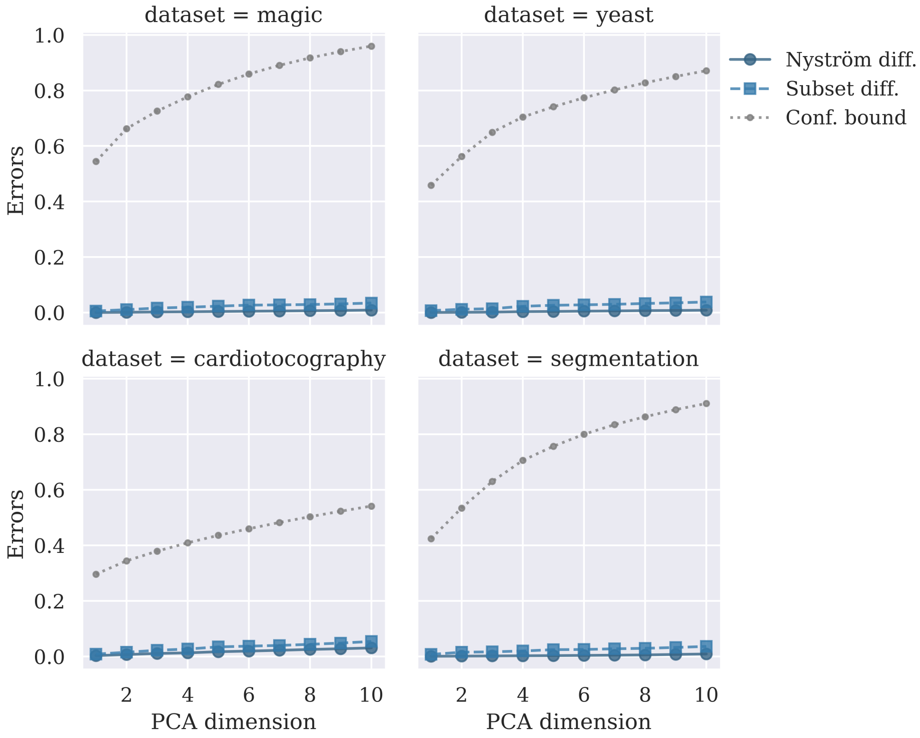


FIGURE 1. Error comparison with the RBF kernel

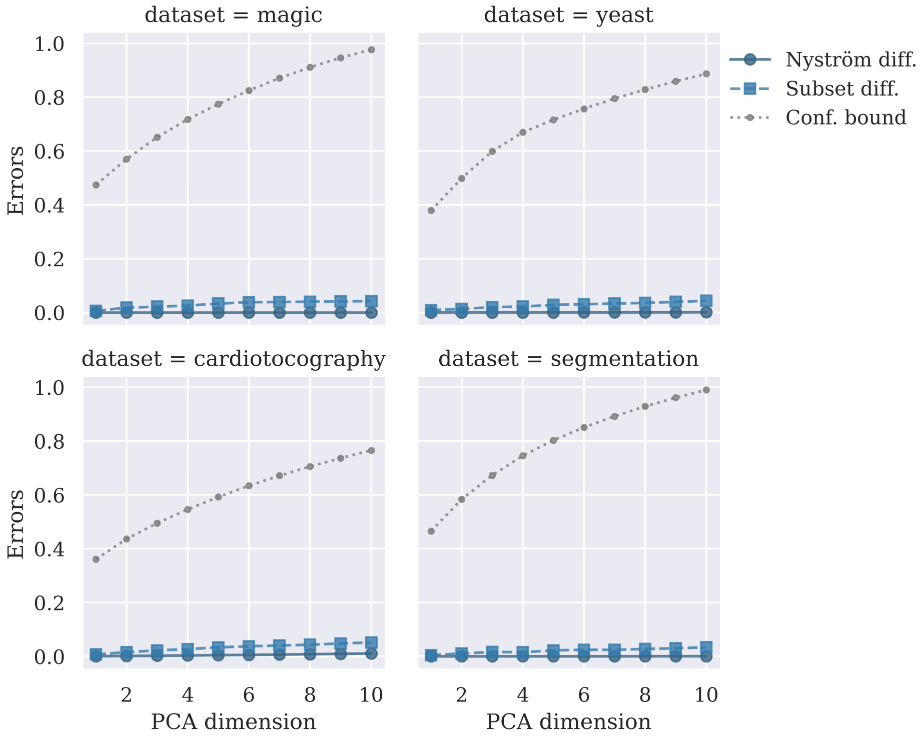


FIGURE 2. Error comparison with the polynomial kernel

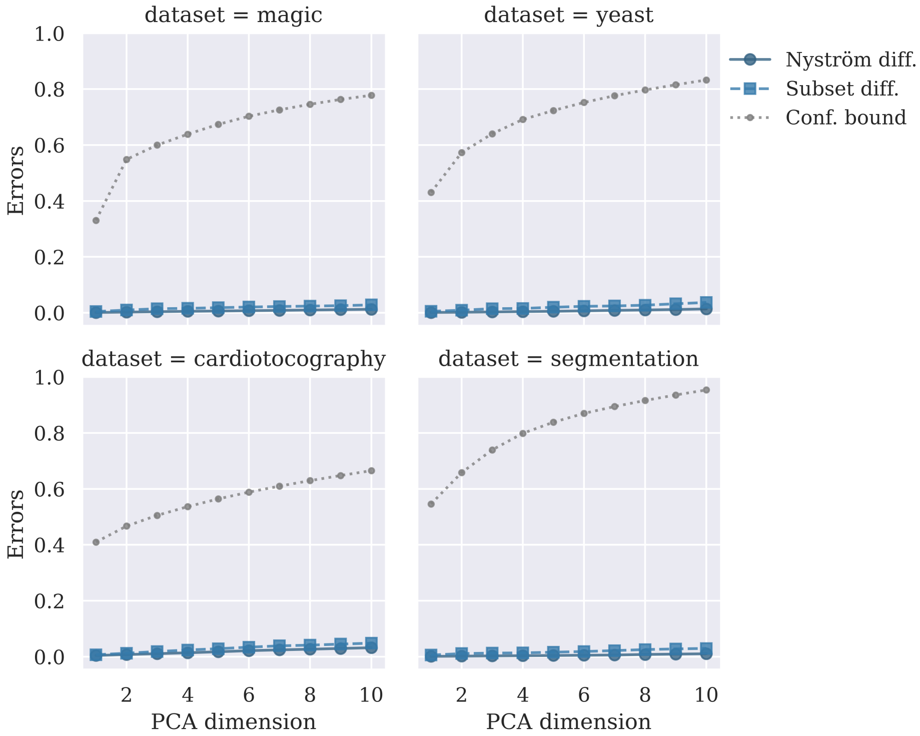


FIGURE 3. Error comparison with the Cauchy kernel

The bound seems fairly conservative for these datasets and these choices of hyperparameters. In real-life applications of the Nyström method the datasets are usually much larger, with the number of data points sometimes in the millions, and with much larger n and m the bound will be significantly smaller. The main purpose of the current experiments is rather to investigate differences between datasets and kernel functions and across PCA dimensions. The same experiments could not be performed for actual datasets where the Nyström method is selected to make kernel PCA scalable, since in this case the calculation of the eigendecomposition is intractable.

8. APPLICATION: NYSTRÖM PRINCIPAL COMPONENT REGRESSION

As an application of Nyström kernel PCA we present kernel principal component regression with the Nyström method, or *Nyström kernel PCR*. The proposed method may be used for regularized kernel regression, for example as an alternative to kernel ridge regression with the Nyström method. Its derivation demonstrates how the principal scores from Nyström kernel PCA may be used as new data points for supervised learning methods.

Principal component regression performs a regression of a target variable onto the principal scores from a subset of the principal components, instead of using the original data as regressor variables [Jolliffe, 2002, Chapter 8]. Principal component regression introduces regularization and ameliorates collinearity of the regressors, which leads to high variances for the coefficient estimates and may especially be a problem for kernel methods. It is known to correspond to the *errors-in-variables* regression model under certain circumstances, where the dependent and independent variables are assumed to contain measurement noise [Fuller, 1980].

We first derive standard kernel PCR without the Nyström method. This derivation appears to be novel, as previous presentations of kernel principal component regression assumed data to have zero mean in feature space [Rosipal et al., 2000, 2001].

Suppose thus that each data point x_i is paired with an observation of a target variable y_i in $\mathbb{R}$ which we wish to predict using a new observation x^* of the independent variable. The regression model is

$$y = \alpha + S_d\beta + \varepsilon$$

with parameters α and $\beta = (\beta_1, \beta_2, ..., \beta_d)^T$, where $y = (y_1, y_2, ..., y_n)^T$, S_d are the principal scores from kernel PCA with respect to the top d principal components, and ε is a noise vector $\varepsilon = (\varepsilon_1, \varepsilon_2, ..., \varepsilon_n)^T$, whose components we assume are generated from a zero-mean distribution with finite variance $\mathrm{Var}(\varepsilon_i)$. The intercept is given by $\alpha = \bar{y}$ since the scores have zero mean in each dimension. From Section 3 the principal scores are given by $S_d = Q_d\Lambda_d^{1/2}$, where $Q_d\Lambda_d Q_d^T$ is the truncated eigendecomposition of K'. Since we assumed Z to be square-integrable we may apply least squares estimation to obtain that [Sen et al., 2010]

$$\hat{\beta} = (S_d^T S_d)^{-1} S_d^T y' = \Lambda_d^{-1/2} Q_d^T y' = S_d^{-1} y'$$

where $y' = (y_1 - \bar{y},\, y_2 - \bar{y},\, ...,\, y_n - \bar{y})^T$. We recall that the principal scores of a new data point x^*, which we centre since we estimated the regression for zero-mean data points, are given by, with respect to the top d principal components

$$w_d^* = \Lambda_d^{-1/2} Q_d^T \kappa'(x^*) = \Lambda_d^{-1/2} Q_d^T \left(\kappa(x^*) - \mathbb{1}_n\kappa(x^*) - K\mathbf{1}_n + \mathbb{1}_n K\mathbf{1}_n\right)$$

and so the prediction for a new data point becomes

$$\hat{y} = \bar{y} + \beta^T w_d^{*T} = \bar{y} + y'^T Q_d \Lambda_d^{-1} Q_d^T \kappa'(x^*)$$

For the Nyström method, the principal scores are given by $W = K'_{nm}K'^{-1/2}_{mm}V = K'_{nm}U$, and so the principal scores with respect to the top d principal components are given by $W_d = K'_{nm}K'^{-1/2}_{mm}V_d = K'_{nm}U_d$ where $V_d\widetilde{\Lambda}_d V_d^T$ is the truncated eigendecomposition of $\frac{1}{n}K'^{-1/2}_{mm}K'_{mn}K'_{nm}K'^{-1/2}_{mm}$ and $U_d = K'^{-1/2}_{mm}V_d$. The regression model then becomes

$$y = \alpha + W_d\beta + \varepsilon = \alpha + K'_{nm}U_d\beta + \varepsilon = \alpha + K'_{nm}K'^{-1/2}_{mm}V_d\beta + \varepsilon$$

The least squares parameter estimates are $\hat{\alpha} = \bar{y}$ and

$$\hat{\beta} = (W_d^T W_d)^{-1} W_d^T y' = \left(V_d^T K'^{-1/2}_{mm} K'_{mn} K'_{nm} K'^{-1/2}_{mm} V_d\right)^{-1} V_d^T K'^{-1/2}_{mm} K'_{mn} y'$$

$$= \left((V_d^T V \widetilde{\Lambda} V^T V_d\right)^{-1} V_d^T K'^{-1/2}_{mm} K'_{mn} y' = \widetilde{\Lambda}_d^{-1} V_d^T K'^{-1/2}_{mm} K'_{mn} y' = \widetilde{\Lambda}_d^{-1} U_d^T K'_{mn} y'$$

And so the prediction becomes

$$\hat{y} = \bar{y} + y'^T K'_{nm} U_d \widetilde{\Lambda}_d^{-1} U_d^T \,\widetilde{\kappa}(x^*)$$

We implement kernel principal component regression with the Nyström method (Nyström KPCR) in computer experiments and compare it with Nyström kernel ridge regression (Nyström KRR) [Rudi et al., 2015], which is given by[7]

$$\hat{y} = \bar{y} + \beta^T \kappa(x^*)$$

$$\hat{\beta} = (K_{mn} K_{nm} + \gamma K_{mm})^{-1} K_{mn} y'$$

where $\gamma \geq 0$ is a regularization parameter.

We use the `airfoil` dataset from the UCI machine learning repository [Dua and Graff, 2017], which describes aerodynamic tests of blades in a wind tunnel from NASA and contains 1503 data points and 6 attributes. Again we normalize the attributes to have mean 0 and variance 1. Note that we must not normalize the entire dataset at once so as to not introduce look-ahead bias in the regression – when creating a prediction for a new data point we need to normalize this data point using the mean and variance from the training set.

We use the radial basis functions kernel with parameter $\sigma = 1$. The source code for these experiments is available in the same package at `https://github.com/fredhallgren/nystrompca`. We estimate the regression on a training dataset with a random sample of 75 % of all data points, and evaluate the method on a test set with the remaining data points.

We plot the R^2 for the regression on the test set for different subset sizes m, ridge parameters γ and PCA dimensions d below in Figure 4. For each parameter combination a different subset is used.

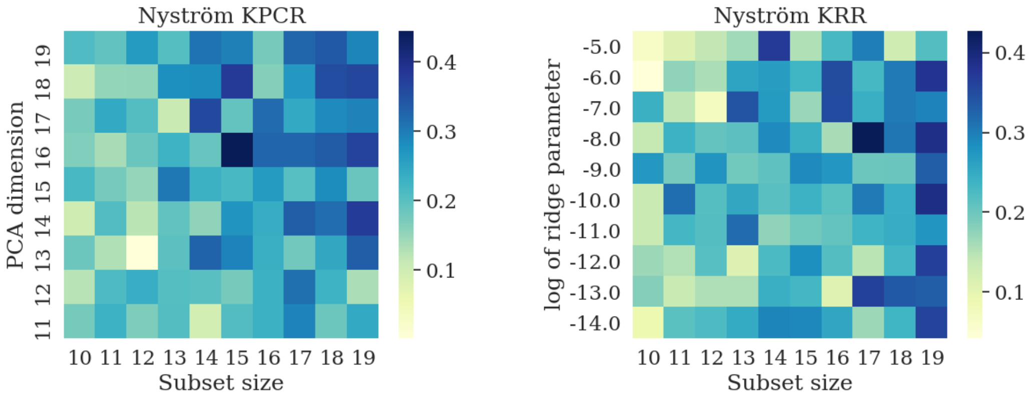


FIGURE 4. Heat maps with regression R^2

For Nyström kernel PCR the regression accuracy improves as we increase the number of principal components used in the regression and as the size of the subset increases. For Nyström KRR

[7] This is a slightly different specification than in Rudi et al. [2015], where we have demeaned the target variable and subsumed a factor n into the ridge parameter

the accuracy also improves with a larger subset, but the pattern is less clear as we change the regularization parameter.

These experiments can also be run with the command-line tool, using the below command

```
> nystrompca regression -m 100 -d 90
```

To further elucidate the behaviour of the methods we also plot the actual target values versus the predicted ones on the test set for one instance of the parameters. Please see below Figure 5. We now use $m = 100$, $d = 90$ and $\gamma = 10^{-11}$. The parameters d and γ were manually tuned. In this particular example Nyström KPCR obtained an R^2 of 0.74 and Nyström KRR 0.72 when we set the seed to 1.

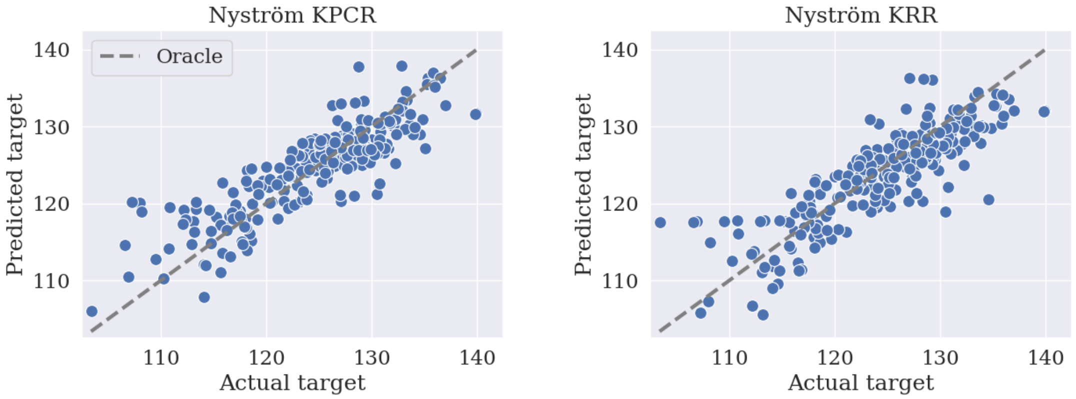


FIGURE 5. Scatter plot with regression predictions

The scatter plots of the predictions versus the actual targets look as expected for an R^2 of around 0.7. The predictions for the two methods look quite similar, but slightly different characteristics are exhibited by the plots due to the different regularization methodologies.

9. CONCLUSION

In this paper we have presented an efficient implementation of non-linear PCA by combining kernel PCA with the Nyström method, providing the principal components, explained variance, the principal scores and the reconstruction error. The algorithm centres the data according to the standard definition of PCA.

We further showed that there is little use in applying the Nyström method from the perspective of the reconstruction error when the number of subsampled data points is equal to the PCA dimension. In this case it is preferable to create the principal components directly from only the subset of data points.

We also provided a finite-sample confidence bound on the empirical reconstruction error of the method, which allows us to measure its statistical accuracy before the entire dataset has been observed. The bound assumes data has zero mean in feature space, but could potentially be adapted to account for centring of data points, although the analysis would become more involved and the notation more unwieldy.

The principal scores from the method may be used instead of the original data matrix in any supervised learning method, in order for example to achieve regularization and denoising. We demonstrated this for linear regression by presenting Nyström kernel principal component regression.

We hope that the work presented in this paper will be of interest to the academic community and to industry practitioners, and that it may give ideas about future directions of research.

In addition to linear regression, there are many other methods based on PCA where kernel PCA with the Nyström method could be analyzed and explored, such as when PCA is applied in discriminant analysis, outlier detection or dictionary learning. The latter could be achieved for example along the lines of Golts and Elad [2016].

The approximate Nyström kernel matrix $\widetilde{K} = K_{nm}K_{mm}^{-1}K_{mn}$ may often be used as a drop-in replacement for the original kernel matrix to speed up kernel machines. However, for many methods, like kernel PCA, more work is needed for a complete treatment. There are still many kernel methods where application of the Nyström method is not necessarily trivial and has not been fully derived.

Kernel PCA is closely related to functional PCA. Functional PCA may also suffer from scalability issues if the individual functions are sampled at a large number of points. It's possible that there are settings where the Nyström method could be successfully applied to functional data analysis for improved computational efficiency.

APPENDIX A. PROOFS

In this section we present the proofs of Propositions 1 and 2, and Theorems 1, 2 and 3.

Proof of Theorem 1. Standard principal component analysis finds the perpendicular intersecting lines in $\mathbb{R}^d$ along which the variance of the data is successively maximized. These lines are affine subspaces of $\mathbb{R}^d$ which are orthogonal with respect to the associated vector space. To derive kernel PCA with the Nyström method we apply PCA in the span of the subset of data points $\mathcal{H}_S$, i.e. finding the orthogonal one-dimensional affine subspaces of $\mathcal{H}_S$ where the projected data has maximum variance. These are on the form

$$\phi_0 + \langle f_j \rangle = \phi_0 + \{\, a f_j \,|\, a \in \mathbb{R} \,\}$$

where $\phi_0 \in \mathcal{H}_S$ is the translation of the vector space $\langle f_j \rangle$, and the $f_j \in \mathcal{H}_S$, taken to have norm one, are the principal components. It is known from standard PCA that the translation vector is given by the mean of the data points, which in our case is the mean of the data points projected onto $\mathcal{H}_S$. Using $P_{\mathcal{H}_S} = G_m^*(G_m G_m^*)^{-1} G_m = m \cdot G_m^* K_{mm}^{-1} G_m$, where G_m is the sampling operator [Rudi et al., 2015], we obtain

$$\phi_0 = \frac{1}{n}\sum_{r=1}^{n} P_{\mathcal{H}_S}\phi(x_r) = \frac{1}{n}\sum_{r=1}^{n} m \cdot G_m^* K_{mm}^{-1} G_m \phi(x_r)$$

$$= \frac{1}{n}\sum_{r=1}^{n} \sqrt{m} \cdot G_m^* K_{mm}^{-1} \kappa_m(x_r) = \frac{1}{n} K_{nm} K_{mm}^{-1} \kappa_m(x)$$

Any element $\phi \in \mathcal{H}_S$ can be written as $\phi = \phi_0 + \sum_{k=1}^{m} a_k \cdot (\phi(x_k) - \phi_0)$ for some coefficients $a_1, a_2, ..., a_m$ and so the principal components are on the form $f_j = \sum_{k=1}^{m} u_{j,k}(\phi(x_k) - \phi_0)$ with coefficients $u_{j,1}, u_{j,2}, ..., u_{j,m}$. The affine projection of a data point $\phi(x)$ onto $\phi_0 + \langle f_j \rangle$ is then

$$P_{\phi_0 + \langle f_j \rangle}\phi(x) = \phi_0 + \langle \phi(x) - \phi_0, f_j \rangle_{\mathcal{H}} f_j$$

The variance of the full dataset along $\phi_0 + \langle f_j \rangle$ then becomes

$$\mathrm{Var}_{f_j}\left(\{\phi(x_i)\}_{i=1}^{n}\right) = \frac{1}{n}\sum_{i=1}^{n}\left(\phi_0 + \langle \phi(x_i) - \phi_0, f_j \rangle_{\mathcal{H}} - \frac{1}{n}\sum_{\ell=1}^{n}\left(\phi_0 + \langle \phi(x_\ell) - \phi_0, f_j \rangle_{\mathcal{H}}\right)\right)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left\langle \phi(x_i) - \frac{1}{n}\sum_{\ell=1}^{n}\phi(x_\ell), \sum_{k=1}^{m} u_{j,k}\left(\phi(x_k) - \phi_0\right)\right\rangle_{\mathcal{H}}^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{k=1}^{m} u_{j,k}\left(k_{k,i} - \frac{1}{n}\sum_{\ell=1}^{n} k_{k,\ell} - \langle \phi(x_i), \phi_0 \rangle_{\mathcal{H}} + \frac{1}{n}\sum_{\ell=1}^{n}\langle \phi(x_\ell), \phi_0 \rangle_{\mathcal{H}}\right)\right)^2$$

Using

$$\langle \phi(x_i), P_{\mathcal{H}_S}\phi(x_r)\rangle_{\mathcal{H}} = \langle \phi(x_i), m \cdot G_m^* K_{mm}^{-1} G_m \phi(x_r)\rangle_{\mathcal{H}}$$

$$= \sqrt{m}\langle \phi(x_i), G_m^* K_{mm}^{-1}\kappa_m(x_r)\rangle_{\mathcal{H}} = \kappa_m(x_i)^T K_{mm}^{-1}\kappa_m(x_r)$$

where $\kappa_m(x) = (k(x_1, x),\ k(x_2, x),\ ...,\ k(x_m, x))^T$, and setting $\kappa_m(x_a) = \kappa_{m,a}$, we obtain

$$\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{k=1}^{m} u_{j,k}\left(k_{k,i} - \frac{1}{n}\sum_{\ell=1}^{n} k_{k,\ell} - \langle \phi(x_i), \phi_0\rangle_{\mathcal{H}} + \frac{1}{n}\sum_{\ell=1}^{n}\langle \phi(x_\ell), \phi_0\rangle_{\mathcal{H}}\right)\right)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{k=1}^{m} u_{j,k}\left(k_{k,i} - \frac{1}{n}\sum_{\ell=1}^{n} k_{k,\ell} - \frac{1}{n}\sum_{r=1}^{n}\kappa_{m,i}^T K_{mm}^{-1}\kappa_{m,r} + \frac{1}{n^2}\sum_{\substack{\ell=1\\ r=1}}^{n}\kappa_{m,\ell}^T K_{mm}^{-1}\kappa_{m,r}\right)\right)^2$$

$$= \frac{1}{n} u_j^T K'_{mn} K'_{nm} u_j$$

where

$$K'_{mn} = K_{mn} - K_{mn}\mathbb{1}_n - \mathbb{1}_n^{m,n} K_{nm} K_{mm}^{-1} K_{mn} + \mathbb{1}_n^{m,n} K_{nm} K_{m,m}^{-1} K_{mn}\mathbb{1}_n$$

with $\mathbb{1}_n^{m,n}$ an $m \times n$ matrix with each element equal to $\frac{1}{n}$, and $K'_{nm} = K'^{T}_{mn}$.

The principal components are then given by the orthonormal vectors $f_j = \sum_{k=1}^{m} u_{j,k}(\phi(x_k) - \phi_0)$, $j = 1, 2, ..., m$ that successively maximize the variance. The inner product between two principal components is

$$\langle f_j, f_p\rangle_{\mathcal{H}} = \left\langle \sum_{k=1}^{m} u_{j,k}\left(\phi(x_k) - \phi_0\right), \sum_{q=1}^{m} u_{p,q}\left(\phi(x_q) - \phi_0\right)\right\rangle_{\mathcal{H}}$$

$$= \sum_{\substack{k=1\\ q=1}}^{m} u_{j,k}u_{p,q}\left(k_{k,q} - \frac{1}{n}\sum_{r=1}^{n}\kappa_{m,r}K_{mm}^{-1}\kappa_{m,k} - \frac{1}{n}\sum_{\ell=1}^{n}\kappa_{m,\ell}K_{mm}^{-1}\kappa_{m,q} + \frac{1}{n^2}\sum_{\substack{r=1\\ \ell=1}}^{n}\kappa_{m,r}K_{mm}^{-1}\kappa_{m,\ell}\right)$$

$$= u_j^T K'_{mm} u_p$$

where $K'_{mm} = K_{mm} - \mathbb{1}_n^{m,n}K_{nm} - K_{mn}\mathbb{1}_n^{n,m} + \mathbb{1}_n^{m,n}K_{nm}K_{mm}^{-1}K_{mn}\mathbb{1}_n^{m,n}$. Maximizing the variance therefore becomes a generalized eigenvalue problem. We have

$$\langle f_j, f_p\rangle_{\mathcal{H}} = u_j^T K'_{mm} u_p = \left(K'^{1/2}_{mm} u_j\right)^T \left(K'^{1/2}_{mm} u_p\right) := v_j^T v_p$$

where $K'^{1/2}_{mm}$ is the unique positive semi-definite square root of K'_{mm} given by $m \cdot U^m \Lambda^{m\,1/2} U^{m\,T}$, where $U^m \Lambda^m U^{m\,T}$ is the eigendecomposition of $\frac{1}{m}K'_{mm}$. Therefore the variance can be written

$$\frac{1}{n} v_j^T K'^{-1/2}_{mm} K'_{mn} K'_{nm} K'^{-1/2}_{mm} v_j = \left\langle v_j,\ \frac{1}{n} K'^{-1/2}_{mm} K'_{mn} K'_{nm} K'^{-1/2}_{mm} v_j\right\rangle_{\mathbb{R}^m}$$

Then by the Courant-Fischer-Weyl theorem [Bhatia, 1997, Corollary III.1.2] the maximum values over successively orthonormal vectors v_j are given by the eigenvalues of $\frac{1}{n}{K'_{mm}}^{-1/2}K'_{mn}K'_{nm}{K'_{mm}}^{-1/2}$, and they occur at its eigenvectors. These eigenvectors will be unique (up to a sign), since all data points are different by assumption.

The principal components are then given by

$$\widetilde{\phi}_j = \sum_{k=1}^{m} u_{j,k}\left(\phi(x_k) - \phi_0\right) \qquad j = 1, 2, ..., m$$

where $u_j = {K'_{mm}}^{-1/2}v_j$, and the affine subspaces with maximum variances are $\{\, \phi_0 + t\widetilde{\phi}_j \mid t \in \mathbb{R}\,\}$, $j = 1, 2, ..., m$.

The principal score of a centred data point i with respect to the principal component j is given by

$$\begin{aligned} w_{j,i} &= \left\langle \phi(x_i) - \frac{1}{n}\sum_{\ell=1}^{n}\phi(x_\ell),\ \sum_{k=1}^{m}u_{j,k}(\phi(x_k) - \phi_0)\right\rangle_{\mathcal{H}} \\ &= \sum_{k=1}^{m}u_{j,k}\left(k_{k,i} - \frac{1}{n}\sum_{\ell=1}^{n}k_{k,\ell} - \frac{1}{n}\sum_{r=1}^{n}\kappa_{m,i}^{T}K_{mm}^{-1}\kappa_{m,r} + \frac{1}{n^2}\sum_{\substack{\ell=1\\ r=1}}^{n}\kappa_{m,\ell}^{T}K_{mm}^{-1}\kappa_{m,r}\right) \end{aligned}$$

for $j = 1, 2, ..., n$. Or in matrix format

$$(w_{i,j}) = W = K'_{nm}U$$

where $U = {K'_{mm}}^{-1/2}V$ and $\frac{1}{n}{K'_{mm}}^{-1/2}K'_{mn}K'_{nm}{K'_{mm}}^{-1/2} = V\widetilde{\Lambda}V^T$, and so $W = K'_{nm}{K'_{mm}}^{-1/2}V$.

The scores of a *new* data point x^* which is centred in feature space, i.e. the coordinates of $\phi(x^*) - \frac{1}{n}\sum_{\ell=1}^{n}\phi(x_\ell)$ in terms of the principal components, are given by

$$\begin{aligned} w_j^* &= \left\langle \phi(x^*) - \frac{1}{n}\sum_{\ell=1}^{n}\phi(x_\ell),\ \sum_{k=1}^{m}u_{j,k}\left(\phi(x_k) - \phi_0\right)\right\rangle_{\mathcal{H}} \\ &= \sum_{k=1}^{m}u_{j,k}\left(k(x_k, x^*) - \frac{1}{n}\sum_{\ell=1}^{n}k_{k,\ell} - \frac{1}{n}\sum_{r=1}^{n}\kappa_{m,r}^{T}K_{mm}^{-1}\kappa_m(x^*) + \frac{1}{n^2}\sum_{\substack{r=1\\ \ell=1}}^{n}\kappa_{m,r}^{T}K_{mm}^{-1}\kappa_{m,\ell}\right) \end{aligned}$$

or in matrix format

$$w^* = U^T\left(\kappa_m(x^*) - K_{mn}\mathbf{1}_n - \mathbb{1}_n^{m,n}K_{nm}K_{mm}^{-1}\kappa_m(x^*) + \mathbb{1}_n^{m,n}K_{nm}K_{mm}^{-1}K_{mn}\mathbf{1}_n\right) := U^T\,\widetilde{\kappa}(x^*)$$

where $\mathbf{1}_n$ is a length-n column vector given by $\mathbf{1}_n = (\frac{1}{n}, \frac{1}{n}, \ ..., \ \frac{1}{n})^T$.

□

Proof of Theorem 2. The projection of a data point $\phi(x_i)$ onto a principal component is given by

$$
\begin{aligned}
P_{\hat{\phi}_j^{m,n}}\phi(x_i) &= \frac{1}{\sqrt{m\hat{\lambda}_j^m}} \sum_{k=1}^{m} u_{j,k}^m \, \langle \phi(x_i),\, \phi(x_k) - \phi_0 \rangle_{\mathcal{H}} \, \hat{\phi}_j^{m,n} \\
&= \frac{1}{\sqrt{m\hat{\lambda}_j^m}} \sum_{k=1}^{m} u_{j,k}^m \left(k(x_k, x_i) - \frac{1}{n} K_{nm} K_{mm}^{-1} \kappa_m(x_i) \right) \hat{\phi}_j^{m,n}
\end{aligned}
$$

where $(\hat{\lambda}_j^m, u_j^m)$ is the jth eigenpair of $\frac{1}{m} K'_{mm}$ and $u_{j,k}^m$ is the kth element of u_j^m [Shawe-Taylor et al., 2005].

The projection of a centred data point $\phi'(x_i)$ is then, similarly to Theorem 1, with $k_{a,b} := k(x_a, x_b)$ and $\kappa_m(x_a) = \kappa_{m,a}$

$$
P_{\hat{\phi}_j^{m,n}}\phi'(x_i) = \frac{1}{\sqrt{m\hat{\lambda}_j^m}} \sum_{k=1}^{m} u_{j,k}^m \left\langle \phi(x_i) - \frac{1}{n} \sum_{\ell=1}^{n} \phi(x_\ell),\, \phi(x_k) - \phi_0 \right\rangle_{\mathcal{H}} \hat{\phi}_j^{m,n}
$$

$$
= \frac{1}{\sqrt{m\hat{\lambda}_j^m}} \sum_{k=1}^{m} u_{j,k}^m \left(k_{k,i} - \frac{1}{n} \sum_{\ell=1}^{n} k_{k,\ell} - \frac{1}{n} \sum_{r=1}^{n} \kappa_{m,i}^T K_{mm}^{-1} \kappa_{m,r} + \frac{1}{n^2} \sum_{\substack{\ell=1 \\ r=1}}^{n} \kappa_{m,\ell}^T K_{mm}^{-1} \kappa_{m,r} \right) \hat{\phi}_j^{m,n}
$$

Taking the norm and summing over $\phi(x_1), \phi(x_2), ..., \phi(x_n)$ we obtain

$$
\frac{1}{n} \sum_{i=1}^{n} \| P_{\hat{\phi}_j^{m,n}} \phi'(x_i) \|_{\mathcal{H}}^2 =
$$

$$
\frac{1}{n \cdot m\hat{\lambda}_j^m} \sum_{i=1}^{n} \left(\sum_{k=1}^{m} u_{j,k}^m \left(k_{k,i} - \frac{1}{n} \sum_{\ell=1}^{n} k_{k,\ell} - \frac{1}{n} \sum_{r=1}^{n} \kappa_{m,i}^T K_{mm}^{-1} \kappa_{m,r} + \frac{1}{n^2} \sum_{\substack{\ell=1 \\ r=1}}^{n} \kappa_{m,\ell}^T K_{mm}^{-1} \kappa_{m,r} \right) \right)^2
$$

$$
= \frac{1}{n \cdot m\hat{\lambda}_j^m} u_j^{m\,T} K'_{mn} K'_{nm} u_j^m \;=:\; \hat{\lambda}_j^{m,n}
$$

For the reconstruction error we have

$$
\begin{aligned}
R_n(\hat{V}_d^m) &= \frac{1}{n} \sum_{i=1}^{n} \| \phi'(x_i) - P_{\hat{V}_d^m} \phi'(x_i) \|_{\mathcal{H}}^2 = \frac{1}{n} \sum_{i=1}^{n} \| \phi'(x_i) \|_{\mathcal{H}} - \frac{1}{n} \sum_{i=1}^{n} \left\| P_{\hat{V}_d^m} \phi'(x_i) \right\|_{\mathcal{H}} \\
&= \frac{1}{n} \mathrm{Tr}(K') - \frac{1}{n} \sum_{i=1}^{n} \left\| P_{\hat{V}_d^m} \phi'(x_i) \right\|_{\mathcal{H}}
\end{aligned}
$$

And so similarly to above, the second term becomes

$$\frac{1}{n}\sum_{i=1}^{n}\left\|P_{\hat{V}_d^m}\phi'(x_i)\right\|_{\mathcal{H}}^2=$$

$$\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{d}\frac{1}{\sqrt{m\hat{\lambda}_j^m}}\sum_{k=1}^{m}u_{j,k}^m\left(k_{k,i}-\frac{1}{n}\sum_{\ell=1}^{n}k_{k,\ell}-\frac{1}{n}\sum_{r=1}^{n}\kappa_{m,i}^T K_{mm}^{-1}\kappa_{m,r}+\frac{1}{n^2}\sum_{\substack{\ell=1\\r=1}}^{n}\kappa_{m,\ell}^T K_{mm}^{-1}\kappa_{m,r}\right)\right)^2$$

$$=\frac{1}{n\cdot m}\mathrm{Tr}(K'_{nm}U_d^m\Lambda_d^{m-1}U_d^{mT}K'_{mn})$$

with $U_d^m\Lambda_d^m U_d^{mT}$ the truncated eigendecomposition of $\frac{1}{m}K'_{mm}$.

□

Proof of Proposition 1. Since $\hat{V}_d^m \subset \mathcal{H}_S$ for any d and by Theorem 1

$$\widetilde{\lambda}_{<d} = \max_{\substack{\dim(V)=d\\ a+V\subset\mathcal{H}_S}} \frac{1}{n}\sum_{i=1}^{n}\|P_{a+V}z_i\|_{\mathcal{H}}^2 = \max_{\substack{\dim(V)=d\\ V\subset\mathcal{H}_S\\ a\in\mathcal{H}_S}} \frac{1}{n}\sum_{i=1}^{n}\|P_V(z_i-a)\|_{\mathcal{H}}^2$$

$$\geq \frac{1}{n}\sum_{i=1}^{n}\|P_{\hat{V}_d^m}(z_i-\phi_0)\|_{\mathcal{H}}^2 = \hat{\lambda}_{<d}^{m,n}$$

The case $d=m$ follows since both $\langle\{\hat{\phi}_j^{m,n}\}_{j=1}^m\rangle$ and $\langle\{\widetilde{\phi}_j\}_{j=1}^m\rangle$ capture the full variance of the data in $\mathcal{H}_S$.

□

Proof of Proposition 2. By the previous proposition we have $\widetilde{V}_m = \hat{V}_m^m$ for a fixed ω and so we will have $\widetilde{V}_m \overset{d}{=} \hat{V}_m^m$ if $\{X_{i_1}, X_{i_2}, ..., X_{i_m}\} \overset{d}{=} \{X_1, X_2, ..., X_m\}$, where $S=\{i_1, i_2, ..., i_m\}$ are the indices for the subsampled data points. By the law of total probability

$$\mathbb{P}(\{X_{i_1}\leq a_1, X_{i_2}\leq a_2,\ ...,\ X_{i_m}\leq a_m\})$$

$$=\sum_S \mathbb{P}(\{X_{i_1}\leq a_1, X_{i_2}\leq a_2,\ ...,\ X_{i_m}\leq a_m\}|S)\mathbb{P}(S)$$

$$=\sum_S \mathbb{P}(\{X_1\leq a_1, X_2\leq a_2,\ ...,\ X_m\leq a_m\}|S)\mathbb{P}(S)$$

since conditional on the sample S, we have m random variables generated according to $\mathbb{P}_X$, which we can take to be $X_1, X_2, ..., X_m$.

If the subsampling is independent of the data then

$$\sum_S \mathbb{P}(\{X_1 \leq a_1, X_2 \leq a_2,\ ...,\ X_m \leq a_m\}|S)\mathbb{P}(S)$$

$$= \mathbb{P}(\{X_1 \leq a_1, X_2 \leq a_2,\ ...,\ X_m \leq a_m\}) \sum_S \mathbb{P}(S) = \prod_{k=1}^{m} \mathbb{P}(\{X_k \leq a_k\})$$

so the subsampled data points are generated i.i.d. from $\mathbb{P}_X$. We can therefore conclude that $\widetilde{V}_m \overset{d}{=} \hat{V}_m^m$. Since Z has the same distribution $\mathbb{P}_Z$ regardless of the subspace and since $\widetilde{V}_m \overset{d}{=} \hat{V}_m^m$ we have $P_{\widetilde{V}_m} Z' \overset{d}{=} P_{\hat{V}_m^m} Z'$ and can conclude that, since Z is square-integrable

$$\mathbb{E}[\|P_{\widetilde{V}_m} Z' - Z'\|_{\mathcal{H}}^2] = \mathbb{E}[\|P_{\hat{V}_m^m} Z' - Z'\|_{\mathcal{H}}^2]$$

and so $R(\widetilde{V}_m) = R(\hat{V}_m^m)$ when $p(S \,|\, x_1, x_2, ..., x_n) = p(S)$.

□

Proof of Theorem 3. The difference in errors can be rewritten through

$$R_n(\widetilde{V}_d) - R_n(\hat{V}_d) = \min_{\substack{\dim(V)=d \\ V \subset \mathcal{H}_S}} \frac{1}{n} \sum_{i=1}^{n} \|P_V z_i - z_i\|_{\mathcal{H}}^2 - \min_{\dim(V)=d} \frac{1}{n} \sum_{i=1}^{n} \|P_V z_i - z_i\|_{\mathcal{H}}^2$$

$$= \max_{\dim(V)=d} \frac{1}{n} \sum_{i=1}^{n} \|P_V z_i\|_{\mathcal{H}}^2 - \max_{\substack{\dim(V)=d \\ V \subset \mathcal{H}_S}} \frac{1}{n} \sum_{i=1}^{n} \|P_V z_i\|_{\mathcal{H}}^2$$

$$= \frac{1}{n} \sum_{i=1}^{n} \|(P_{\mathcal{H}_S} + P_{\mathcal{H}_S^\perp}) P_{\hat{V}_d} z_i\|_{\mathcal{H}}^2 - \max_{\substack{\dim(V)=d \\ V \subset \mathcal{H}_S}} \frac{1}{n} \sum_{i=1}^{n} \|P_V z_i\|_{\mathcal{H}}^2$$

$$\leq \frac{1}{n} \sum_{i=1}^{n} \|P_{\mathcal{H}_S} P_{\hat{V}_d} z_i\|_{\mathcal{H}}^2 + \frac{1}{n} \sum_{i=1}^{n} \|P_{\mathcal{H}_S^\perp} P_{\hat{V}_d} z_i\|_{\mathcal{H}}^2 - \max_{\substack{\dim(V)=d \\ V \subset \mathcal{H}_S}} \frac{1}{n} \sum_{i=1}^{n} \|P_V z_i\|_{\mathcal{H}}^2$$

$$\leq \frac{1}{n} \sum_{i=1}^{n} \|P_{\mathcal{H}_S^\perp} P_{\hat{V}_d} z_i\|_{\mathcal{H}}^2$$

Expanding the projection operator $P_{\hat{V}_d}$ we obtain

$$\frac{1}{n}\sum_{i=1}^{n}\|P_{\mathcal{H}_S^\perp}P_{\hat{V}_d}z_i\|_{\mathcal{H}_S}^2 = \frac{1}{n}\sum_{i=1}^{n}\left\|P_{\mathcal{H}_S^\perp}\sum_{j=1}^{d}\langle z_i,\hat{\phi}_j^n\rangle_{\mathcal{H}}\hat{\phi}_j^n\right\|_{\mathcal{H}}^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left\|\sum_{j=1}^{d}\langle z_i,\hat{\phi}_j^n\rangle_{\mathcal{H}}P_{\mathcal{H}_S^\perp}\hat{\phi}_j^n\right\|_{\mathcal{H}}^2 \leq \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{d}\left\|\langle z_i,\hat{\phi}_j^n\rangle_{\mathcal{H}}P_{\mathcal{H}_S^\perp}\hat{\phi}_j^n\right\|_{\mathcal{H}}^2$$

The last inequality is fairly tight. It becomes an equality without the projection $P_{\mathcal{H}_S^\perp}$, and the further the projection is from the identity, the smaller the norm of $P_{\mathcal{H}_S^\perp}\hat{\phi}_j^n$. Now we have

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{d}\left\|\langle z_i,\hat{\phi}_j^n\rangle_{\mathcal{H}}P_{\mathcal{H}_S^\perp}\hat{\phi}_j^n\right\|_{\mathcal{H}}^2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{d}|\langle z_i,\hat{\phi}_j^n\rangle_{\mathcal{H}}|^2\left\|P_{\mathcal{H}_S^\perp}\hat{\phi}_j^n\right\|_{\mathcal{H}}^2$$

$$= \sum_{j=1}^{d}\left(\frac{1}{n}\sum_{i=1}^{n}|\langle z_i,\hat{\phi}_j^n\rangle_{\mathcal{H}}|^2\right)\left\|P_{\mathcal{H}_S^\perp}\hat{\phi}_j^n\right\|_{\mathcal{H}}^2 = \sum_{j=1}^{d}\hat{\lambda}_j^n\left\|P_{\mathcal{H}_S^\perp}\hat{\phi}_j^n\right\|_{\mathcal{H}}^2$$

Expanding the other projection operator we get

$$\sum_{j=1}^{d}\hat{\lambda}_j^n\left\|P_{\mathcal{H}_S^\perp}\hat{\phi}_j^n\right\|_{\mathcal{H}}^2 = \sum_{j=1}^{d}\hat{\lambda}_j^n\left\|\hat{\phi}_j^n - P_{\mathcal{H}_S}\hat{\phi}_j^n\right\|_{\mathcal{H}}^2 = \sum_{j=1}^{d}\hat{\lambda}_j^n\left\|\hat{\phi}_j^n - \sum_{k=1}^{m}\langle\hat{\phi}_j^n,\hat{\phi}_k^m\rangle_{\mathcal{H}}\hat{\phi}_k^m\right\|_{\mathcal{H}}^2$$

We may only keep the jth index in the sum over the m data points without losing much accuracy

$$\sum_{j=1}^{d}\hat{\lambda}_j^n\left\|\hat{\phi}_j^n - \sum_{k=1}^{m}\langle\hat{\phi}_j^n,\hat{\phi}_k^m\rangle_{\mathcal{H}}\hat{\phi}_k^m\right\|_{\mathcal{H}}^2 \leq \sum_{j=1}^{d}\hat{\lambda}_j^n\left\|\hat{\phi}_j^n - \langle\hat{\phi}_j^n,\hat{\phi}_j^m\rangle_{\mathcal{H}}\hat{\phi}_j^m\right\|_{\mathcal{H}}^2 = \sum_{j=1}^{d}\hat{\lambda}_j^n\left(1-\langle\hat{\phi}_j^n,\hat{\phi}_j^m\rangle_{\mathcal{H}}^2\right)$$

Since $\cos\theta = \langle\hat{\phi}_j^n,\hat{\phi}_j^m\rangle_{\mathcal{H}}$ then by the Davis-Kahan $\sin 2\theta$ theorem [Davis and Kahan, 1970]

$$1-\langle\hat{\phi}_j^n,\hat{\phi}_j^m\rangle_{\mathcal{H}}^2 = \sin^2\theta \leq \frac{\|C_n - C_m\|_{\mathrm{HS}(\mathcal{H})}^2}{\left(\hat{\lambda}_j^m - \hat{\lambda}_{j+1}^m\right)^2}$$

We know that the left-hand side is always less than or equal to 1 so we have

$$1-\langle\hat{\phi}_j^n,\hat{\phi}_j^m\rangle_{\mathcal{H}}^2 \leq \frac{\|C_n - C_m\|_{\mathrm{HS}(\mathcal{H})}^2}{\left(\hat{\lambda}_j^m - \hat{\lambda}_{j+1}^m\right)^2} \wedge 1$$

Next we have, by Lidskii's inequality [Kato, 2013, Chapter 3, Theorem 6.11]

$$\sum_{j=1}^{d} \hat{\lambda}_j^n \left(\frac{\|C_n - C_m\|^2_{\mathrm{HS}(\mathcal{H})}}{\left(\hat{\lambda}_j^m - \hat{\lambda}_{j+1}^m\right)^2} \wedge 1 \right) = \sum_{j=1}^{d} \left(\hat{\lambda}_j^m + \hat{\lambda}_j^n - \hat{\lambda}_j^m\right) \left(\frac{\|C_n - C_m\|^2_{\mathrm{HS}(\mathcal{H})}}{\left(\hat{\lambda}_j^m - \hat{\lambda}_{j+1}^m\right)^2} \wedge 1 \right)$$

$$\leq \sum_{j=1}^{d} \hat{\lambda}_j^m \left(\frac{\|C_n - C_m\|^2_{\mathrm{HS}(\mathcal{H})}}{\left(\hat{\lambda}_j^m - \hat{\lambda}_{j+1}^m\right)^2} \wedge 1 \right) + \sum_{j=1}^{d} \left| \left(\hat{\lambda}_j^n - \hat{\lambda}_j^m\right) \right| \max_{1 \leq k \leq d} \frac{\|C_n - C_m\|^2_{\mathrm{HS}(\mathcal{H})}}{\left(\hat{\lambda}_k^m - \hat{\lambda}_{k+1}^m\right)^2} \wedge 1$$

$$\leq \sum_{j=1}^{d} \hat{\lambda}_j^m \left(\frac{\|C_n - C_m\|^2_{\mathrm{HS}(\mathcal{H})}}{\left(\hat{\lambda}_j^m - \hat{\lambda}_{j+1}^m\right)^2} \wedge 1 \right) + \|C_n - C_m\|_{\mathrm{HS}(\mathcal{H})} \max_{1 \leq k \leq d} \frac{\|C_n - C_m\|^2_{\mathrm{HS}(\mathcal{H})}}{\left(\hat{\lambda}_k^m - \hat{\lambda}_{k+1}^m\right)^2} \wedge 1$$

Now the only unknown and random quantity is $\|C_n - C_m\|_{\mathrm{HS}(\mathcal{H})}$. It depends both on the unobserved data points $z_{m+1}, z_{m+2}, ..., z_n$ and the observed ones $z_1, z_2, ..., z_m$. First we rewrite it into one term that only contains observed data points and another that only contains unobserved ones

$$\begin{aligned}
\|C_n - C_m\|_{\mathrm{HS}(\mathcal{H})} &= \left\| \frac{1}{n} \sum_{i=1}^{n} \otimes^2 z_i - \frac{1}{m} \sum_{r=1}^{m} \otimes^2 z_r \right\|_{\mathcal{H} \otimes \mathcal{H}} \\
&= \left\| \frac{1}{n} \sum_{i=m+1}^{n} \otimes^2 z_i - \frac{n-m}{nm} \sum_{r=1}^{m} \otimes^2 z_r \right\|_{\mathcal{H} \otimes \mathcal{H}} \\
&= \frac{n-m}{n} \left\| \frac{1}{n-m} \sum_{i=m+1}^{n} \otimes^2 z_i - \frac{1}{m} \sum_{r=1}^{m} \otimes^2 z_r \right\|_{\mathcal{H} \otimes \mathcal{H}} \\
&= \frac{n-m}{n} \|C_{n-m} - C_m\|_{\mathrm{HS}(\mathcal{H})}
\end{aligned}$$

Noting that $\mathbb{E}[C_m] = \mathbb{E}\left[\frac{1}{m}\sum_{r=1}^{m} \otimes^2 z_r\right] = \frac{1}{m}\sum_{r=1}^{m} \mathbb{E}\left[\otimes^2 z_r\right] = \mathbb{E}\left[\otimes^2 Z\right] = \frac{1}{n-m}\sum_{i=m+1}^{n} \mathbb{E}\left[\otimes^2 z_i\right] = \mathbb{E}[C_{n-m}]$ we may split the norm up into two separate independent norms

$$\begin{aligned}
&\frac{n-m}{n} \|C_{n-m} - C_m\|_{\mathrm{HS}(\mathcal{H})} \\
&\leq \frac{n-m}{n} \left(\|C_{n-m} - \mathbb{E}[C_{n-m}]\|_{\mathrm{HS}(\mathcal{H})} + \|C_m - \mathbb{E}[C_m]\|_{\mathrm{HS}(\mathcal{H})} \right)
\end{aligned} \tag{6}$$

If we let $Y_i = \otimes^2 z_i - \mathbb{E}[\otimes^2 Z]$, then the random variables Y_i have zero mean, and they are bounded by $B := \sup_x k(x, x)$ since both $\otimes^2 z_i$ and $\mathbb{E}[\otimes^2 Z]$ are positive. This can be seen for example as follows.

Consider the Hilbert subspace of $\mathcal{H} \otimes \mathcal{H}$ of positive operators, denoted $(\mathcal{H} \otimes \mathcal{H})_+$, which is closed and so indeed a Hilbert space, and let $L_1, L_2, T \in (\mathcal{H} \otimes \mathcal{H})_+$. We recall that by the Riesz representation theorem every Hilbert space can be identified with its dual through the isometry $L_1 \mapsto \langle \cdot, L_1 \rangle$. And so $\|L_1 - L_2\|_{\mathrm{HS}(\mathcal{H})} = \|f_1 - f_2\|$ for some bounded linear functionals f_1, f_2 in $(\mathcal{H} \otimes \mathcal{H})^*$.

To see that each f_1, f_2 will be positive, note that $f_1(T) = \langle T, L_1 \rangle_{\mathrm{HS}(\mathcal{H})} = \sum_{i=1}^{\infty} \langle Te_i, L_1 e_i \rangle_{\mathcal{H}}$ for any basis $\{e_i\}$ in $\mathcal{H}$. If we take $\{e_i\}$ to be the eigenvectors of L_1, arbitrarily extended to a basis for the entire space if $\mathrm{Ker}(L_1) \neq \{0\}$, we obtain, for each i, that $\langle Te_i, L_1 e_i \rangle_{\mathcal{H}} = \langle Te_i, \lambda_i e_i \rangle_{\mathcal{H}} = \lambda_i \langle Te_i, e_i \rangle_{\mathcal{H}} \geq 0$ since T is positive and $\lambda_i \geq 0$. And so $f_1(T) \geq 0$ for each T.

Since f_1, f_2 are positive everywhere we have $\|f_1 - f_2\| \leq \max\{\|f_1\|, \|f_2\|\} \leq B$.

Then by Hoeffding's inequality in Banach spaces [Pinelis, 1994, Theorem 3.5], we have that with confidence $1 - 2e^{-\delta}$

$$\frac{n-m}{n} \left\| \frac{1}{n-m} \sum_{i=m+1}^{n} \otimes^2 z_i - \mathbb{E}[\otimes^2 Z] \right\|_{\mathcal{H}\otimes\mathcal{H}} \leq \frac{n-m}{n} \frac{\sqrt{2\delta}B}{\sqrt{n-m}} = \sqrt{2\delta}B \frac{\sqrt{n-m}}{n}$$

In the second term above in Equation (6) the data points are observed but the expectation is unknown. Through an application of the evaluation operator we can still use Hoeffding's inequality to devise a bound as follows. We have, since $E_\omega(a) = a$ if a is constant and since $E_\omega(Z_i) = E_\omega(Z_j)$ even if $i \neq j$

$$\left\| \frac{1}{m} \sum_{r=1}^{m} \otimes^2 z_r - \mathbb{E}[\otimes^2 Z] \right\|_{\mathcal{H}\otimes\mathcal{H}} = \left\| \frac{1}{m} \sum_{r=1}^{m} E_{\omega_r}(\otimes^2 Z_r) - \mathbb{E}[\otimes^2 Z] \right\|_{\mathcal{H}\otimes\mathcal{H}}$$

$$= \left\| \frac{1}{m} \sum_{r=1}^{m} E_{\omega_r}\left(\otimes^2 Z_r - \mathbb{E}[\otimes^2 Z]\right) \right\|_{\mathcal{H}\otimes\mathcal{H}} = \left\| \frac{1}{m^2} \sum_{r=1}^{m} \sum_{\ell=1}^{m} E_{\omega_r}\left(\otimes^2 Z_\ell - \mathbb{E}[\otimes^2 Z]\right) \right\|_{\mathcal{H}\otimes\mathcal{H}}$$

$$= \left\| \frac{1}{m} \left(\sum_{r=1}^{m} E_{\omega_r} \right) \left(\frac{1}{m} \sum_{\ell=1}^{m} \otimes^2 Z_\ell - \mathbb{E}[\otimes^2 Z] \right) \right\|_{\mathcal{H}\otimes\mathcal{H}} \leq \frac{1}{m} \left\| \sum_{r=1}^{m} E_{\omega_r} \right\| \left\| \frac{1}{m} \sum_{r=1}^{m} \otimes^2 Z_r - \mathbb{E}[\otimes^2 Z] \right\|_1$$

$$\leq \frac{1}{m} \sum_{r=1}^{m} \|E_{\omega_r}\| \left\| \frac{1}{m} \sum_{r=1}^{m} \otimes^2 Z_r - \mathbb{E}[\otimes^2 Z] \right\|_1 = B \cdot \left\| \frac{1}{m} \sum_{r=1}^{m} \otimes^2 Z_r - \mathbb{E}[\otimes^2 Z] \right\|_1$$

$$= B \cdot \mathbb{E} \left[\left\| \frac{1}{m} \sum_{r=1}^{m} \otimes^2 Z_r - \mathbb{E}[\otimes^2 Z] \right\|_{\mathcal{H}\otimes\mathcal{H}} \right]$$

Through another application of Hoeffding's inequality we obtain that

$$\mathbb{P}\left(\left\| \frac{1}{m} \sum_{r=1}^{m} \otimes^2 Z_r - \mathbb{E}[\otimes^2 Z] \right\|_{\mathcal{H}\otimes\mathcal{H}} \leq \frac{\sqrt{2\delta}B}{\sqrt{m}} \right) \geq 1 - 2e^{-\delta}$$

We can then bound the distribution function of $\left\|\frac{1}{m}\sum_{r=1}^m \otimes^2 Z_r - \mathbb{E}[\otimes^2 Z]\right\|_{\mathcal{H}\otimes\mathcal{H}}$ through

$$\mathbb{P}\left(\left\|\frac{1}{m}\sum_{r=1}^m \otimes^2 Z_r - \mathbb{E}[\otimes^2 Z]\right\|_{\mathcal{H}\otimes\mathcal{H}} \leq y\right) \geq \left(1 - 2\exp\left\{-\frac{y^2 m}{2B^2}\right\}\right) \vee 0 =: F(y)$$

Letting $\sigma^2 := B^2/m$ we obtain

$$\mathbb{E}\left[\left\|\frac{1}{m}\sum_{r=1}^m \otimes^2 Z_r - \mathbb{E}[\otimes^2 Z]\right\|_{\mathcal{H}\otimes\mathcal{H}}\right] \leq \int_{\mathbb{R}_+} (1 - F(y))dy$$

$$= \sigma\sqrt{2\log 2} + \int_{\sigma\sqrt{2\log 2}}^{+\infty} 2e^{-y^2/2\sigma^2} dy = \sigma\sqrt{2\log 2} + 2\sqrt{2\pi}\sigma\frac{1}{\sqrt{2\pi}\sigma}\int_{\sigma\sqrt{2\log 2}}^{+\infty} e^{-y^2/2\sigma^2} dy$$

$$= \sigma\sqrt{2\log 2} + 2\sqrt{2\pi}\sigma\Phi\left(-\sqrt{2\log 2}\right)$$

Adding together the two terms gives that with confidence $1 - 2e^{-\delta}$

$$\|C_n - C_m\|_{\mathrm{HS}(\mathcal{H})} = \frac{n-m}{n}\left(B\frac{\sqrt{2\delta}}{\sqrt{n-m}} + \frac{B^2}{\sqrt{m}}\left(\sqrt{2\log 2} + 2\sqrt{2\pi}\,\Phi\left(-\sqrt{2\log 2}\right)\right)\right) =: D$$

and so also with confidence $1 - 2e^{-\delta}$ that

$$R_n(\widetilde{V}_d) - R_n(\hat{V}_d) \leq \sum_{j=1}^d \hat{\lambda}_j^m \left(\frac{D^2}{\left(\hat{\lambda}_j^m - \hat{\lambda}_{j+1}^m\right)^2} \wedge 1\right) + D \cdot \max_{1\leq k\leq d} \frac{D^2}{\left(\hat{\lambda}_k^m - \hat{\lambda}_{k+1}^m\right)^2} \wedge 1$$

We recall that the eigenvalues of the empirical covariance operator equal the eigenvalues of the kernel matrix $\frac{1}{m}K_{mm}$, which completes the proof.

□

References

M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1710377.

F. Bach. On the equivalence between kernel quadrature rules and random feature expansions. *The Journal of Machine Learning Research*, 18(1):714–751, 2017. http://www.jmlr.org/papers/volume18/15-178/15-178.pdf.

F. R. Bach and M. I. Jordan. Kernel independent component analysis. *The Journal of Machine Learning Research*, 3(Jul):1–48, 2002. http://www.jmlr.org/papers/volume3/bach02a/bach02a.pdf.

S. Banach. Théorie des opérations linéaires. *Monografie Matematyczne*, 1932. http://kielich.amu.edu.pl/Stefan_Banach/pdf/teoria-operacji-fr/banach-teorie-des-operations-lineaires.pdf.

A. Beaulieu. *Learning SQL: Generate, Manipulate, and Retrieve Data*. O'Reilly Media, 3rd edition, 2020. https://www.oreilly.com/library/view/learning-sql-3rd/9781492057604/.

Y. Bengio, O. Delalleau, N. L. Roux, J.-F. Paiement, P. Vincent, and M. Ouimet. Learning eigenfunctions links spectral embedding and kernel PCA. *Neural Computation*, 16(10):2197–2219, 2004. https://www.mitpressjournals.org/doi/pdfplus/10.1162/0899766041732396.

P. Besse. Approximation spline de l'analyse en composantes principales d'une variable aléatoire hilbertienne. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, volume 12, pages 329–349. Université Paul Sabatier, 1991. https://afst.centre-mersenne.org/article/AFST_1991_5_12_3_329_0.pdf.

P. Besse and J. O. Ramsay. Principal components analysis of sampled functions. *Psychometrika*, 51(2): 285–311, 1986. https://link.springer.com/content/pdf/10.1007/BF02293986.pdf.

R. Bhatia. *Matrix analysis*, volume 169 of *Graduate texts in mathematics*. Springer Science & Business Media, 1st edition, 1997. https://link.springer.com/book/10.1007/978-1-4612-0653-8.

G. Blanchard, O. Bousquet, and L. Zwald. Statistical properties of kernel principal component analysis. *Machine Learning*, 66(2-3):259–294, 2007. https://link.springer.com/content/pdf/10.1007/s10994-006-6895-9.pdf.

B. Bollobás. *Linear analysis*. Cambridge mathematical textbooks. Cambridge University Press, 2nd edition, 1999. https://www.cambridge.org/core/books/linear-analysis/E43EE4282F2D8636117A47A4F110E8FE.

L. Carratino, S. Vigogna, D. Calandriello, and L. Rosasco. Park: Sound and efficient kernel ridge regression by feature space partitions. *arXiv preprint arXiv:2106.12231*, 2021. https://arxiv.org/pdf/2106.12231.pdf.

D. L. Cohn. *Measure theory*, volume 165 of *Birkhäuser Advanced Texts Basler Lehrbucher*. Springer Science & Business Media, 2nd edition, 1980. https://link.springer.com/book/10.1007/978-1-4614-6956-8.

J. Dauxois, A. Pousse, and Y. Romain. Asymptotic theory for the principal component analysis of a vector random function: some applications to statistical inference. *Journal of Multivariate Analysis*, 12(1):136–154, 1982. https://core.ac.uk/download/pdf/82501258.pdf.

E. B. Davies. *Linear operators and their spectra*, volume 106 of *Cambridge studies in advanced mathematics*. Cambridge University Press, 1st edition, 2007.
https://www.cambridge.org/core/books/linear-operators-and-their-spectra/6DDA33D1D7032F9EBB41194F33C18A69.

C. Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.
https://epubs.siam.org/doi/pdf/10.1137/0707001.

A. C. Davison and D. V. Hinkley. *Bootstrap methods and their application*. Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, 1st edition, 1997.
https://www.cambridge.org/core/books/bootstrap-methods-and-their-application/ED2FD043579F27952363566DC09CBD6A.

D. Dua and C. Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

W. A. Fuller. Properties of some estimators for the errors-in-variables model. *The Annals of Statistics*, pages 407–422, 1980.
https://projecteuclid.org/download/pdf_1/euclid.aos/1176344961.

A. Gittens and M. W. Mahoney. Revisiting the Nyström method for improved large-scale machine learning. *The Journal of Machine Learning Research*, 17(1):3977–4041, 2016.
http://www.jmlr.org/papers/volume17/gittens16a/gittens16a.pdf.

A. Golts and M. Elad. Linearized kernel dictionary learning. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):726–739, 2016.
https://arxiv.org/pdf/1509.05634.pdf.

G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins University Press, 4th edition, 2013.
https://jhupbooks.press.jhu.edu/title/matrix-computations.

C. Graham and D. Talay. *Simulation stochastique et méthodes de Monte-Carlo*. École Polytechnique, Département de Mathématiques Appliquées, 2011.
https://hal.archives-ouvertes.fr/hal-00602795.

V. Guigue, A. Rakotomamonjy, and S. Canu. Kernel basis pursuit. In *European Conference on Machine Learning*, pages 146–157. Springer, 2005.
https://link.springer.com/content/pdf/10.1007/11564096_18.pdf.

M. Haddouche, B. Guedj, O. Rivasplata, and J. Shawe-Taylor. Upper and lower bounds on the performance of kernel PCA. *arXiv preprint arXiv:2012.10369*, 2020.
https://arxiv.org/pdf/2012.10369.pdf.

P. Hall and M. Hosseini-Nasab. On properties of functional principal components analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):109–126, 2006.
https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9868.2005.00535.x.

P. Hall, H.-G. Müller, and J.-L. Wang. Properties of principal component methods for functional and longitudinal data analysis. *The Annals of Statistics*, pages 1493–1517, 2006.
https://projecteuclid.org/download/pdfview_1/euclid.aos/1152540756.

M. C. Hout, M. H. Papesh, and S. D. Goldinger. Multidimensional scaling. *Wiley Interdisciplinary Reviews: Cognitive Science*, 4(1):93–103, 2013.
https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcs.1203.

A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.
https://www.sciencedirect.com/science/article/pii/S0893608000000265.

A. Iosifidis and M. Gabbouj. Nyström-based approximate kernel subspace learning. *Pattern Recognition*, 57:190–197, 2016.
https://www.sciencedirect.com/science/article/pii/S0031320316300036.

I. T. Jolliffe. *Principal component analysis*. Springer Science & Business Media, 2nd edition, 2002.
http://cda.psych.uiuc.edu/statistical_learning_course/Jolliffe%20I.%20Principal%20Component%20Analysis%20(2ed.,%20Springer,%202002)(518s)_MVsa_.pdf.

I. T. Jolliffe and J. Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065): 20150202, 2016.
https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.2015.0202.

T. Kato. *Perturbation theory for linear operators*, volume 132 of *Classics in mathematics*. 2013.
https://link.springer.com/book/10.1007/978-3-642-66282-9.

V. Koltchinskii and E. Giné. Random matrix approximation of spectra of integral operators. *Bernoulli*, 6 (1):113–167, 2000.
https://projecteuclid.org/download/pdf_1/euclid.bj/1082665383.

E. Kreyszig. *Introductory functional analysis with applications*. Wiley, 1st edition, 1989.
http://www-personal.acfr.usyd.edu.au/spns/cdm/resources/Kreyszig%20-%20Introductory%20Functional%20Analysis%20with%20Applications.pdf.

M. Ledoux and M. Talagrand. *Probability in Banach spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.
https://www.springer.com/gp/book/9783642202117.

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *The Journal of Machine Learning Research*, 2(Feb):419–444, 2002.
http://www.jmlr.org/papers/volume2/lodhi02a/lodhi02a.pdf.

S. Ma and M. Belkin. Diving into the shallows: a computational perspective on large-scale shallow learning. In *Advances in Neural Information Processing Systems*, pages 3778–3787, 2017.
https://papers.nips.cc/paper/6968-diving-into-the-shallows-a-computational-perspective-on-large-scale-shallow-learning.pdf.

C. McDiarmid. On the method of bounded differences. *Surveys in Combinatorics*, 141(1):148–188, 1989.
https://www.cambridge.org/no/academic/subjects/mathematics/discrete-mathematics-information-theory-and-coding/surveys-combinatorics-1989-invited-papers-twelfth-british-combinatorial-conference?format=PB&isbn=9780521378239.

G. Meanti, L. Carratino, L. Rosasco, and A. Rudi. Kernel methods through the roof: handling billions of points efficiently. *arXiv preprint arXiv:2006.10350*, 2020.
https://arxiv.org/abs/2006.10350.

S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (cat. no. 98th8468)*, pages 41–48. IEEE, 1999.
https://ieeexplore.ieee.org/abstract/document/788121.

E. J. Nyström. Über die praktische auflösung von integralgleichungen mit anwendungen auf randwertaufgaben. *Acta Mathematica*, 54(1):185–204, 1930.
https://link.springer.com/content/pdf/10.1007/BF02547521.pdf.

V. I. Paulsen and M. Raghupathi. *An introduction to the theory of reproducing kernel Hilbert spaces*, volume 152 of *Cambridge studies in advanced mathematics*. Cambridge University Press, 2016.
https://www.cambridge.org/core/books/an-introduction-to-the-theory-of-reproducing-kernel-hilbert-spaces/C3FD9DF5F5C21693DD4ED812B531269A.

K. Pearson. Principal components analysis. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 6(2):559, 1901.

I. Pinelis. Optimum bounds for the distributions of martingales in Banach spaces. *The Annals of Probability*, pages 1679–1706, 1994.
https://projecteuclid.org/download/pdf_1/euclid.aop/1176988477.

K. M. Ramachandran and C. P. Tsokos. *Mathematical statistics with applications in R*. Academic Press, 2nd edition, 2015.
https://www.elsevier.com/books/mathematical-statistics-with-applications-in-r/ramachandran/978-0-12-417113-8.

C. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2nd edition, 2004.
https://www.springer.com/gp/book/9780387212395.

L. Rosasco, M. Belkin, and E. D. Vito. On learning with integral operators. *The Journal of Machine Learning Research*, 11(Feb):905–934, 2010.
http://www.jmlr.org/papers/volume11/rosasco10a/rosasco10a.pdf.

R. Rosipal, L. J. Trejo, and A. Cichocki. *Kernel principal component regression with EM approach to nonlinear principal components extraction*. University of Paisley, 2000.
http://aiolos.um.savba.sk/~roman/Papers/tr00_2.pdf.

R. Rosipal, M. Girolami, L. J. Trejo, and A. Cichocki. Kernel PCA for feature extraction and de-noising in nonlinear regression. *Neural Computing & Applications*, 10(3):231–243, 2001.
https://www.researchgate.net/profile/Leonard-Trejo/publication/243134486_Kernel_PCA_for_Feature_Extraction_and_De-Noising_in_Nonlinear_Regression/links/583f5da508ae8e63e6182cbf/Kernel-PCA-for-Feature-Extraction-and-De-Noising-in-Nonlinear-Regression.pdf.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
https://science.sciencemag.org/content/290/5500/2323.full.

A. Rudi, R. Camoriano, and L. Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems*, pages 1657–1665, 2015.
https://arxiv.org/pdf/1507.04717.pdf.

A. Rudi, L. Carratino, and L. Rosasco. Falkon: An optimal large scale kernel method. In *Advances in Neural Information Processing Systems*, pages 3888–3898, 2017.
http://papers.nips.cc/paper/6978-falkon-an-optimal-large-scale-kernel-method.pdf.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.
https://www.mitpressjournals.org/doi/pdfplus/10.1162/089976698300017467.

P. K. Sen, J. M. Singer, and A. C. P. De Lima. *From finite sample to asymptotic methods in statistics*, volume 28 of *Cambridge series in statistical and probabilistic mathematics*. Cambridge University Press, 2010.
https://www.cambridge.org/core/books/from-finite-sample-to-asymptotic-methods-in-statistics/07DFA2860E18EDB1A9FE1FF3B4E07F0C.

J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 1st edition, 2004.
https://www.cambridge.org/core/books/kernel-methods-for-pattern-analysis/811462F4D6CD6A536A05127319A8935A.

J. Shawe-Taylor, C. K. Williams, N. Cristianini, and J. Kandola. On the eigenspectrum of the Gram matrix and its relationship to the operator eigenspectrum. In *International Conference on Algorithmic Learning Theory*, pages 23–40. Springer Science & Business Media, 2002.
https://link.springer.com/content/pdf/10.1007%2F3-540-36169-3.pdf.

J. Shawe-Taylor, C. K. Williams, N. Cristianini, and J. Kandola. On the eigenspectrum of the Gram matrix and the generalization error of kernel PCA. *IEEE Transactions on Information Theory*, 51(7): 2510–2522, 2005.
https://homepages.inf.ed.ac.uk/ckiw/postscript/gram.pdf.

R. Singh, M. Sahani, and A. Gretton. Kernel instrumental variable regression. In *Advances in Neural Information Processing Systems*, pages 4593–4605, 2019.
http://papers.neurips.cc/paper/8708-kernel-instrumental-variable-regression.pdf.

M. Sipser. *Introduction to the theory of computation*. Cengage Learning, 3rd edition, 2013.
https://www.cengagebrain.co.uk/shop/isbn/9780357670583.

N. Sterge and B. Sriperumbudur. Statistical optimality and computational efficiency of Nyström kernel PCA. *arXiv preprint arXiv:2105.08875v1*, 2021.
https://arxiv.org/pdf/2105.08875v1.

N. Sterge, B. Sriperumbudur, L. Rosasco, and A. Rudi. Gain with no pain: efficiency of kernel-PCA by Nyström sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 3642–3652. PMLR, 2020.
http://proceedings.mlr.press/v108/sterge20a/sterge20a.pdf.

P. Vincent and Y. Bengio. Kernel matching pursuit. *Machine learning*, 48(1):165–187, 2002.
https://link.springer.com/content/pdf/10.1023/A:1013955821559.pdf.

S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt. Graph kernels. *The Journal of Machine Learning Research*, 11:1201–1242, 2010.
http://www.jmlr.org/papers/volume11/vishwanathan10a/vishwanathan10a.pdf.

T. Wang, Q. Berthet, and R. J. Samworth. Statistical and computational trade-offs in estimation of sparse principal components. *The Annals of Statistics*, 44(5):1896–1930, 2016.
https://projecteuclid.org/download/pdfview_1/euclid.aos/1473685263.

C. K. Williams. On a connection between kernel PCA and metric multidimensional scaling. *Machine Learning*, 46(1):11–19, 2002.
https://link.springer.com/content/pdf/10.1023/A:1012485807823.pdf.

C. K. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, pages 682–688, 2001.
http://papers.nips.cc/paper/1866-using-the-nystrom-method-to-speed-up-kernel-machines.pdf.

S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1-3):37–52, 1987.
http://pzs.dstu.dp.ua/DataMining/pca/bibl/Principal%20components%20analysis.pdf.