Signal-to-Noise Ratio in reproducing kernel Hilbert spaces *

Luis Gómez-Chova[1], Raúl Santos-Rodríguez[2], Gustau Camps-Valls[1]
[1] Image Processing Laboratory (IPL), Universitat de València, Spain.
[2] Department of Engineering Mathematics, University of Bristol, UK.

Abstract

This paper introduces the kernel signal-to-noise ratio (kSNR) for different machine learning and signal processing applications. The kSNR seeks to maximize the signal variance while minimizing the estimated noise variance *explicitly* in a reproducing kernel Hilbert space (rkHs). The kSNR gives rise to considering complex signal-to-noise relations beyond additive noise models, and can be seen as a useful signal-to-noise regularizer for feature extraction and dimensionality reduction. We show that the kSNR generalizes kernel PCA (and other spectral dimensionality reduction methods), least squares SVM, and kernel ridge regression to deal with cases where signal and noise cannot be assumed independent. We give computationally efficient alternatives based on reduced-rank Nyström and projection on random Fourier features approximations, and analyze the bounds of performance and its stability. We illustrate the method through different examples, including nonlinear regression, nonlinear classification in channel equalization, nonlinear feature extraction from high-dimensional spectral satellite images, and bivariate causal inference. Experimental results show that the proposed kSNR yields more accurate solutions and extracts more noise-free features when compared to standard approaches.

Keywords: kernel methods, noise model, signal-to-noise ratio, SNR, heteroscedastic, feature extraction, signal classification, causal inference.

*Paper published in Pattern Recognition Letters 112 (2018) 75–82, doi: 10.1016/j.patrec.2018.06.004

1 Introduction

The signal-to-noise ratio (SNR) describes the proportion of signal power with regard to the noise power, which is an extremely useful concept for quantifying the robustness and quality of a system. In order to reduce the noise, one can try to control the acquisition environment or alternatively look at the noise characteristics and filter the acquired signal accordingly. In fact, several signal processing and machine learning tasks, e.g. filter design or regularization, are linked to the maximization of the SNR, as this enforces smoothness by discarding features influenced by noise while preserving signal characteristics.

In this scenario, a common approach is to transform the observed signal aiming to maximize the SNR, or alternatively minimizing the amount of noise. The minimum noise fraction (MNF) transformation [1] maximizes the variance of the signal and, at the same time, minimizes the estimated noise. However, MNF is a linear transformation that struggles with settings where signal is correlated with the noise (also known as heteroscedastic noise scenarios). MNF assumes an additive noise model and solves a generalized eigenvalue problem taking into account signal and noise covariances, hence no cross-covariance is used. Nevertheless, a more important drawback of the MNF/SNR transformation is that the method cannot deal with nonlinear signal-to-noise relations. To cope with this problem, kernel MNF (kMNF) was presented in [2] for dimensionality reduction. Originally, given the right kernel function, the signal and the estimated noise are mapped to a high-dimensional (feature) space, where the MNF is minimized. This implicit kMNF/kSNR was limited

to feature extraction only, and heavily relied on an accurate noise estimation in the original space.

[3] extended the standard formulation by studying both signal and noise directly in the feature space (*explicit* kMNF). In this way, kSNR can effectively express nonlinear relations between signal and noise and, at the same time, reduces the number of parameters needed [4, 5]. Later, we introduced in [6] the main ideas for exploiting the presented kSNR in machine learning applications beyond feature extraction. In this paper, we analyze the methodology, both theoretically and experimentally. In particular, we note that the kernel version of SNR allows us to consider signal-to-noise dependencies beyond additive noise models, and can be seen as a powerful signal-to-noise regularizer in many applications of machine learning and data processing. It can be applied in combination with any kernel method working under correlated (even non-Gaussian) noise sources. Therefore, we here showcase the *explicit* kSNR for feature extraction, as well as for regression, classification, and bivariate causal inference. Noting the important role of noise estimation, we pay special attention to both *implicit* and *explicit* ways of doing so in both structured and unstructured domains, and relate this to traditional delta tests in multivariate statistics. We complete the theoretical analysis by proposing two alternative formulations to reduce the computational cost involved in the proposed method based on reduced-rank approximations and projections on random Fourier features. Experimentally, we successfully test the method in nonlinear regression problems under different noise sources, causal inference under non-additive noise settings, channel equalization, and nonlinear feature extraction from hyperspectral satellite images.

Section 2 presents the kSNR framework and the specific formulation for the aforementioned problems. Section 3 proposes alternatives to reduce the computational cost of the proposed method and analyzes the stability of the framework. Section 4 presents the experimental results in different applications to show the capabilities of the method. Finally, conclusions are presented in Section 5.

2 Kernel signal-to-noise ratio

In this section, we first introduce the common notation for the nonlinear extensions of the kSNR. In particular, the kernel signal-to-noise ratio is presented in three different contexts: kernel feature extraction, least-squares regression and classification. Finally, the problem of noise estimation is discussed: critical assumptions are made in standard noise estimation in the input space, hence an explicit kernel-based estimation in reproducing kernel Hilbert spaces (rkHs) is introduced.

2.1 kSNR notation

Given a set of training samples $\mathcal{X} := \{\mathbf{x}_i \in \mathbb{R}^d \mid i = 1, ..., n\}$, we assume an additive noise model, $\mathbf{x}_i = \mathbf{s}_i + \mathbf{n}_i$, where the signal is noted as $\mathbf{s}_i$, and the noise $\mathbf{n}_i$ may not necessarily follow a normal distribution. In matrix notation, we can represent observations as $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_n]^\top \in \mathbb{R}^{n \times d}$, where $^\top$ denotes matrix transposition, being typically the number of training samples n higher than the data dimensionality d. $\mathbf{X}$ can be also expressed as the sum of a signal $\mathbf{S}$ and a noise $\mathbf{N}$ matrices, $\mathbf{X} = \mathbf{S} + \mathbf{N}$. The centered version of $\mathbf{X}$ is indicated by $\tilde{\mathbf{X}}$, and the empirical covariance of the observations and noise are calculated as $\mathbf{C}_x = \frac{1}{n}\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}}$ and $\mathbf{C}_n = \frac{1}{n}\tilde{\mathbf{N}}^\top\tilde{\mathbf{N}}$. The noise is commonly assumed to be orthogonal (uncorrelated) to signal, $\mathbf{S}^\top\mathbf{N} = \mathbf{N}^\top\mathbf{S} = \mathbf{0}$, which is very convenient for solving signal-to-noise transformation and blind-source separation problems [1]. The linear MNF/SNR feature extraction is interested in projections most driven by signal and simultaneously less affected by noise. To extract p linear features we project data onto the subspace characterized by the projection matrix $\mathbf{V}$, of size $d \times p$, with $p \leqslant d$, so that data projected onto the top p components are given by $\tilde{\mathbf{X}}' = \tilde{\mathbf{X}}\mathbf{V}$. For extracting only one feature, this problem can be solved by maximizing the so called *Rayleigh quotient*, $(\mathbf{v}^\top\mathbf{C}_x\mathbf{v})/(\mathbf{v}^\top\mathbf{C}_n\mathbf{v})$, which measures the ratio between the desired information and the undesired noise along the direction of $\mathbf{v}$. For extracting more than one feature one could solve the *trace ratio* problem $\mathrm{Tr}\{\mathbf{V}^\top\mathbf{C}_x\mathbf{V}\}/\mathrm{Tr}\{\mathbf{V}^\top\mathbf{C}_n\mathbf{V}\}$, where $\mathrm{Tr}\{\cdot\}$ denotes the trace of a matrix, which is

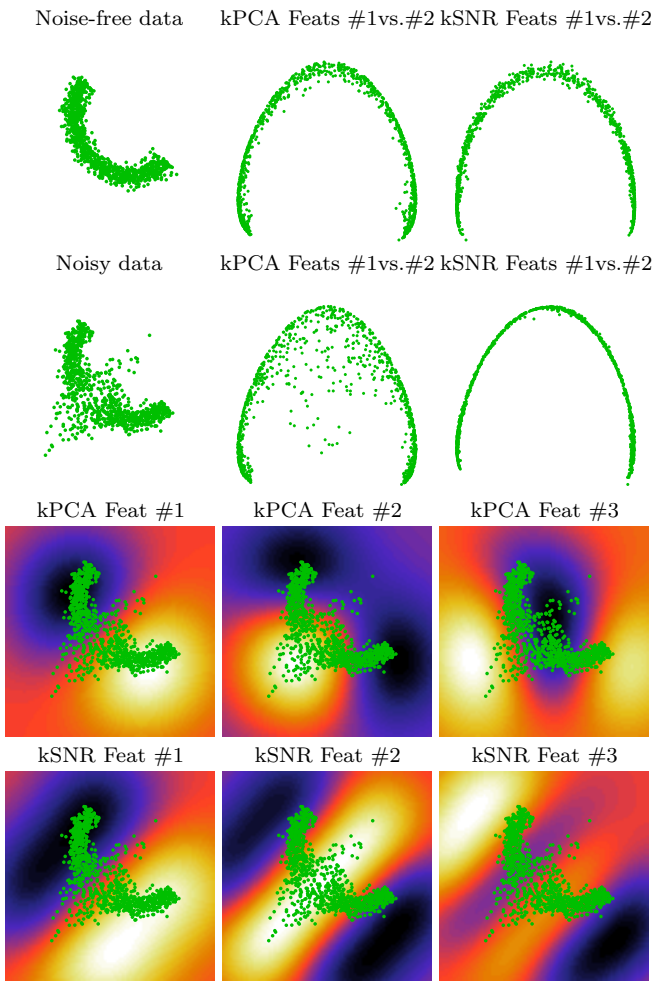

Figure 1: kSNR feature extraction in a two-dimensional example. [*Left panel*] Correlated noise in the $\pi/4$-direction is added to the dataset: in the noise-free case (top), the kSNR is equivalent to kPCA, while in the noisy case (bottom) kPCA projections are affected by noise while kSNR are not. [*Right panel*] Projected features for the noisy dataset help to understand this effect: kPCA projections #2 and #3 capture the noise distribution while, for the kSNR, all extracted projections are invariant to variations in the $\pi/4$ direction where the noise is mostly present, i.e. kSNR avoids projections more affected by noise.

the sum of its eigenvalues and also the cumulative variance of the projected dimensions. However, the *trace ratio* problem does not have a direct closed-form global optimum solution [7] and it is conventionally approximated [1, 2] by solving the associated *ratio trace* problem $\mathrm{Tr}\{(\mathbf{V}^\top \mathbf{C}_n \mathbf{V})^{-1}(\mathbf{V}^\top \mathbf{C}_x \mathbf{V})\}$, which can be stated as:

$$\begin{array}{lll} \text{MNF/SNR} & \text{maximize:} & \mathrm{Tr}\{\mathbf{V}^\top \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \mathbf{V}\} \\ & \text{subject to:} & \mathbf{V}^\top \tilde{\mathbf{N}}^\top \tilde{\mathbf{N}} \mathbf{V} = \mathbf{I}. \end{array} \quad (1)$$

Using Lagrange multipliers to solve this constrained maximization problem shows that the solution (i.e.

the columns of $\mathbf{V}$) is given by the generalized eigenvectors $\mathbf{v}_i$ associated to the largest generalized eigenvalues λ_i of the generalized eigenvalue problem with the signal, $\mathbf{C}_x$, and noise, $\mathbf{C}_n$, covariance matrices:

$$\mathbf{C}_x\mathbf{v}_i = \lambda_i\mathbf{C}_n\mathbf{v}_i. \tag{2}$$

It is important to note for the following discussions that the MNF/SNR transformation (1) does not consider signat-to-noise cross-covariance, and (2) cannot cope with nonlinear feature relations.

In this context, kernel methods allow us to obtain nonlinear extensions of linear problems [8, 9]. The observations $\mathbf{x}_i$ are mapped to a Hilbert space $\mathcal{H}$ via a mapping function $\boldsymbol{\phi}(\cdot)$ that yields high dimensional vectors $\boldsymbol{\phi}(\mathbf{x}_i) \in \mathbb{R}^{d_\mathcal{H}} \subseteq \mathcal{H}$. However, we do not need a direct access to these mapped vectors in order to calculate the dot product between samples in $\mathcal{H}$. It can be computed by using reproducing kernel functions, $K(\mathbf{x}_i, \mathbf{x}_j) = \langle\boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)\rangle_\mathcal{H}$. The squared exponential kernel is typically used in this setting, $K(\mathbf{x}_i,\mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2))$, where σ parameter is the width of this Radial Basis Function (RBF) kernel. The evaluations of the kernel function among all training samples are stored in the kernel matrix $\mathbf{K}$, whose entries are $K(\mathbf{x}_i, \mathbf{x}_j)$.

2.2 kSNR feature extraction

Our interest is to maximize the SNR in Hilbert spaces, which is equivalent to minimize the noise fraction in $\mathcal{H}$:

$$\begin{aligned} \text{kSNR} \qquad & \text{maximize:} \quad \operatorname{Tr}\{\mathbf{U}^\top\tilde{\boldsymbol{\Phi}}^\top\tilde{\boldsymbol{\Phi}}\mathbf{U}\} \\ & \text{subject to:} \quad \mathbf{U}^\top\tilde{\boldsymbol{\Phi}}_n^\top\tilde{\boldsymbol{\Phi}}_n\mathbf{U} = \mathbf{I}, \end{aligned} \tag{3}$$

where $\tilde{\boldsymbol{\Phi}}$ and $\tilde{\boldsymbol{\Phi}}_n \in \mathbb{R}^{n\times d_\mathcal{H}}$ are the matrices containing the centered mapped data and noise samples respectively, and $\mathbf{U}$ is the projection matrix in $\mathcal{H}$ of size $d_\mathcal{H} \times p$. However, this problem is not directly solvable since we do not have access to the mapped samples in $\mathcal{H}$ and its dimension $d_\mathcal{H}$ might be infinite. Making use of the representer's theorem [10] we express the projection matrix as a linear combination of the mapped samples, $\mathbf{U} = \tilde{\boldsymbol{\Phi}}^\top\mathbf{A}$, thus reducing the maximization problem to find the matrix $\mathbf{A}$ of size $n \times p$:

$$\begin{aligned} \text{kSNR} \qquad & \text{maximize:} \quad \operatorname{Tr}\{\mathbf{A}^\top\tilde{\mathbf{K}}^2\mathbf{A}\} \\ & \text{subject to:} \quad \mathbf{A}^\top\tilde{\mathbf{K}}_{xn}\tilde{\mathbf{K}}_{nx}\mathbf{A} = \mathbf{I}, \end{aligned} \tag{4}$$

which is efficiently solved by the generalized eigenproblem:

$$\tilde{\mathbf{K}}^2\boldsymbol{\alpha}_i = \lambda_i\tilde{\mathbf{K}}_{xn}\tilde{\mathbf{K}}_{nx}\boldsymbol{\alpha}_i. \tag{5}$$

This method was proposed in [2] and further extended in [3] for an *explicit* definition of SNR relations in reproducing kernel Hilbert spaces (cf. Section 2.4).

Figure 1 shows the performance of kSNR compared to kPCA in an illustrative example. The proposed kSNR concentrates on extracting components driven by signal and less affected by noise. Interestingly, even though it is not imposed in the signal model, the method considers signal-to-noise nonlinear relations implicitly, and hence can deal with heteroscedastic processes.

Finally, it is worth noting that the kSNR feature extraction generalizes kPCA to cases of non-independent noise. Note that when the noise components are independent in $\mathcal{H}$, $\boldsymbol{\Sigma}_n = \sigma_n^2\mathbf{I}$, then the solution in (5) reduces to the standard kPCA equation, $\tilde{\mathbf{K}}\boldsymbol{\alpha}_i = \lambda_i\boldsymbol{\alpha}_i$.

2.3 kSNR regression and classification

Kernel-based regression and classification problems can also benefit from the maximization of SNR ratios in Hilbert spaces. Let us reformulate standard least squares problems using kernels: the kernel ridge regression (KRR) [9] and least squares SVM (LS-SVM) [11]. In both cases we aim to include the noise covariance matrix in $\mathcal{H}$ as a powerful regularizer. The intuitive idea here is to avoid high variance of the weights in the directions mostly affected by noise. Notationally, the model is given by $\mathbf{y} = \boldsymbol{\Phi}\mathbf{w} + \mathbf{b}$, where $\boldsymbol{\Phi}$ is the matrix of mapped samples, $\boldsymbol{\Phi} := [\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), \ldots, \boldsymbol{\phi}(\mathbf{x}_n)]^\top \in \mathbb{R}^{n\times d_\mathcal{H}}$. The regularized squared loss function to minimize is

$$\min_{\mathbf{w}} \left\{ \|\mathbf{y} - \boldsymbol{\Phi}\mathbf{w}\|^2 + \lambda\mathbf{w}^\top\boldsymbol{\Sigma}_n\mathbf{w} \right\}, \tag{6}$$

where λ is the regularization parameter, the noise covariance in Hilbert space is $\boldsymbol{\Sigma}_n = \boldsymbol{\Phi}_n^\top \boldsymbol{\Phi}_n \in \mathbb{R}^{d_\mathcal{H} \times d_\mathcal{H}}$, and $\boldsymbol{\Phi}_n$ is a matrix containing the estimated noise samples mapped to $\mathcal{H}$, $\boldsymbol{\Phi}_n := [\boldsymbol{\phi}(\mathbf{n}_1) \cdots \boldsymbol{\phi}(\mathbf{n}_n)]^\top \in \mathbb{R}^{n \times d_\mathcal{H}}$.

Hereafter, we intentionally drop the bias term $\mathbf{b}$ for simplicity, even though it was taken into account in all applications. Taking derivatives with respect to $\mathbf{w}$ and applying the representer's theorem [10], $\mathbf{w} = \boldsymbol{\Phi}^\top \boldsymbol{\alpha}$, we obtain the solution expressed as a function of the (dual) weights in $\boldsymbol{\alpha}$:

$$\boldsymbol{\alpha} = (\mathbf{K}^2 + \lambda \mathbf{K}_{xn} \mathbf{K}_{nx})^{-1} \mathbf{K} \mathbf{y}, \tag{7}$$

where $\mathbf{K}_{xn}$ contains the similarities between observations and their estimated noise, i.e. $\mathbf{K}_{xn} = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{n}_j) \rangle_\mathcal{H}$. We can write the solution as $\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{K}^{-1} \mathbf{K}_{xn} \mathbf{K}_{nx})^{-1} \mathbf{y}$ alternatively. Therefore, the term $\boldsymbol{\Omega} := \mathbf{K}^{-1} \mathbf{K}_{xn} \mathbf{K}_{nx}$ can be interpreted as a regularizer that intuitively discounts the impact of noisy samples, and reinforces the importance of the noise-free ones. This essentially goes in the line of discovering relevant directions in feature spaces mainly governed by signal and less affected by noise [12, 13] (cf. Fig. 1).

The kSNR regression model can be used for testing on new incoming examples $\mathbf{X}_*$: we only need to map them to feature spaces, $\boldsymbol{\Phi}_*$ and project them onto the solution vector $\mathbf{w}$. This leads to the predictions $\hat{\mathbf{y}}_* = \boldsymbol{\Phi}_* \mathbf{w} = \boldsymbol{\Phi}_* \boldsymbol{\Phi}^\top \boldsymbol{\alpha} = \mathbf{K}_* \boldsymbol{\alpha}$, where matrix $\mathbf{K}_*$ estimates the similarities between all test and training examples. Note that in the test phase, noise estimation (cf. section 2.4) is not necessary either, since its information is implicitly in model weights.

It is also interesting to note that the kSNR regression generalizes KRR to cases of non-independent noise. For independent noise in Hilbert space, $\boldsymbol{\Sigma}_n = \sigma_n^2 \mathbf{I}$, the solution (7) reduces to the standard KRR, $\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y}$, and λ is related to the noise power σ_n^2. Off-diagonal entries in $\boldsymbol{\Omega}$ stand out and account for signal-to-noise feature relations not accounted when assuming signal and noise to be independent.

The least squares SVM classification model [11][ch. 03] equivalently considers the signal model $f(\mathbf{x}_i) = \text{sign}(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b)$, and introduces equality constraints $\mathbf{y}_i(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b) = 1 - \mathbf{e}_i$, where $\mathbf{e}_i$ represent the residuals (slacks). The kSNR for classification is thus equivalent to the KRR model. Both for regression and classification, the model solution is not sparse (all training examples are accounted for the solution). Nevertheless, the kSNR regularizer seeks for sparsity in feature spaces assigning higher weights to noise-free samples than to noisy ones. This interestingly allows us to generalize the kernel Fisher's discriminant analysis to cope with correlated (possibly nonlinear) signal-to-noise relations [12].

2.4 Noise estimation

kSNR formulation requires estimating the sample noise, which can be a difficult task. In audio and image processing and time series analysis, the most common approach consists in assuming locally stationary signals that allow to estimate the noise as a simple difference between observations, $\hat{\mathbf{n}}_i \approx \mathbf{x}_i - \mathbf{x}_{i-1}$. Other more elaborated approaches approximate the observed signal using autoregressive models to describe the local relations in structured domains. Good examples of these local relations are previous values in a time series or close pixels in an image, which allow to estimate the noise as $\hat{\mathbf{n}}_i \approx \mathbf{x}_i - \sum_{l \in W_i} a_l \mathbf{x}_l$, denoting W_i the neighborhood for the sample $\mathbf{x}_i$. In problems in which there is not a clear structured domain, it is possible to calculate k-nearest neighbors (k-NN) estimates of the noise $\hat{\mathbf{n}}_i \approx \mathbf{x}_i - 1/k \sum_{l \in C} \mathbf{x}_l$, denoting $C = \{1, \ldots, k\}$ the set of k neighbors of $\mathbf{x}_i$. This simple way of noise estimation goes in the line of the *delta test*, which was proposed for time series analysis in [14], and intuitively seeks to estimate the residuals support.

The proposed noise estimation strategies in the input space $\mathcal{X}$ are intuitive and straightforward but present a clear drawback: the two kernels required in the kSNR formulation, $\mathbf{K}$ and $\mathbf{K}_{xn}$, deal with conceptually different objects (observations and noise). Therefore, estimating the noise in the input space (*implicit* kSNR) implies choosing different kernel parameters for $\mathbf{K}$ and $\mathbf{K}_{xn}$. Moreover, the signal-to-noise kernel $\mathbf{K}_{xn}$ handles entities that can be really different in nature and magnitude, which makes the selection of the kernel parameters much more difficult. In fact, by using different kernel parameters for

$\mathbf{K}$ and $\mathbf{K}_{xn}$, one is mapping signal and noise to different Hilbert spaces. In this case, one cannot assume that the eigenvalues obtained with the kSNR transformation have the meaning of SNR in $\mathcal{H}$ anymore. In order to address this problem, we propose what we call *explicit* kSNR, in which the noise is estimated *explicitly* in $\mathcal{H}$ [3]. Basically, as we do in the input space, we encode previous knowledge about the problem to estimate the noise in $\mathcal{H}$ in terms of the mapped samples, $\hat{\boldsymbol{\phi}}(\mathbf{n}_i) \approx \boldsymbol{\phi}(\mathbf{x}_i) - \sum_l a_l \boldsymbol{\phi}(\mathbf{x}_l)$. Therefore, the dot product $\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{n}_j) \rangle_{\mathcal{H}}$ gives rise to the *explicit* signal-to-noise kernel function

$$K_{xn}(\mathbf{x}_i, \mathbf{n}_j) \approx K(\mathbf{x}_i, \mathbf{x}_j) - \sum_l a_l K(\mathbf{x}_i, \mathbf{x}_l), \qquad (8)$$

which can be directly used in the solutions obtained in (5) and (7). Although the performance of the method will depend on the adopted kernel, this approach allows more robust noise estimation in the kernel space, since it may not be always guaranteed that close samples in the input space are also close when mapped to the rkHs. The main rationale behind this approach is that, if neighbors are used as a smoothing in the original space, we should follow the same principle in the transformed space. As mentioned before, the noise estimation coefficients a_l are given by the particular problem in structured domains or by the k-NN approximation in unstructured domains when no additional information is available.

The *explicit* kSNR formulation presents obvious benefits: 1) The hyperparameters of the signal and noise kernel functions are the same since now $\mathbf{K}_{xn}$ is also expressed in terms of similarities between samples in the input space $\mathcal{X}$; 2) The eigenvalues obtained by the *explicit* kSNR transformation can be interpreted as data variance and also as the SNR in the projected space since data and noise are computed in the same Hilbert space; and 3) Using non-linear kernels in kSNR (Eq. (5)) not only allows to extract projections that account for higher order signal and noise relations but in turn introduces (through cross-kernels $\mathbf{K}_{xn}$ and $\mathbf{K}_{nx}$) the cross-covariance between signal and noise in the Hilbert space. This allows to treat problems of signal-dependent noise sources (such as heteroscedastic noise) and thus extends the standard assumption of additive noise to more general signal-to-noise relations. However it is worth noting that the choice of a suitable kernel for a given noise reduction problem is still an open question.

3 Computational efficiency and stability

One of the main shortcomings of kSNR is related to the computational cost since several $n \times n$ kernel matrices are involved. For example, while the standard SNR algorithm for feature extraction has a cost of $\mathcal{O}(d^3)$, our kernel counterparts scale cubically with the number of samples, $\mathcal{O}(n^3)$. Here we propose two alternatives to speed up kSNR. We give the derivation for the particular case of feature extraction, yet similar derivations can be readily obtained for the other developments. In addition, the stability of the obtained solution can be always a problem when solving a generalized eigenproblem using a finite number of samples.

3.1 Reduced-rank kSNR

Besides the high computational cost involved in the previous formulations, model solutions are not generally sparse, so application to new data requires the evaluation of n kernel functions *per* test example, becoming prohibitive for large n. In order to alleviate this problem we propose an alternative low-rank version of the kSNR by reducing the representation space. Let us now consider a reduced rank expansion $\mathbf{U} = \tilde{\boldsymbol{\Phi}}_r^\top \mathbf{A}$ in r vectors rather than all available n training points in Eq. (3). Let us denote $\tilde{\mathbf{K}}_{rx} = \boldsymbol{\Phi}_r \boldsymbol{\Phi}^\top$ and $\mathbf{K}_{rn} = \boldsymbol{\Phi}_r \boldsymbol{\Phi}_n^\top$, where $\boldsymbol{\Phi}_r$ is a subset of the training data containing r samples ($r \ll n$). Now signal and noise covariance matrices in Hilbert spaces can be estimated with only r points, which ultimately lead to the reduced-rank kSNR (RR-kSNR) problem

$$\tilde{\mathbf{K}}_{rx} \tilde{\mathbf{K}}_{xr} \boldsymbol{\alpha}_i = \lambda_i \tilde{\mathbf{K}}_{rn} \tilde{\mathbf{K}}_{nr} \boldsymbol{\alpha}_i, \qquad (9)$$

which involves a generalized eigenproblem with smaller matrices of size $r \times r$, and hence its computational cost only is $\mathcal{O}(r^3)$, $r \ll n$. We want to highlight

here that this is not a simple subsampling, because the model considers correlations between all training data and the reduced subset through $\tilde{\mathbf{K}}_{rx}$. This particular Nyström approximation yields also important advantages in storage and in prediction time. Figure 2(a) shows the evolution of the computational cost as a function of r in a toy example.

3.2 Randomized kSNR

An outstanding result in the kernel methods literature makes use of a classical definition in harmonic analysis to improve approximation and scalability [15]. The Bochner's theorem states that a continuous kernel $K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x} - \mathbf{x}')$ on $\mathbb{R}^d$ is positive definite (p.d.) if and only if K is the Fourier transform of a non-negative measure. If a shift-invariant kernel K is properly scaled, its Fourier transform $p(\mathbf{w})$ is a proper probability distribution. This property is used to approximate kernel functions and matrices with linear projections on a number of D random features, as follows:

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}') &= \int_{\mathbb{R}^d} p(\mathbf{w}) e^{-\mathbb{i}\mathbf{w}^\top(\mathbf{x}-\mathbf{x}')} \mathrm{d}\mathbf{w} \\ &\approx \sum\nolimits_{i=1}^{D} \tfrac{1}{D} e^{-\mathbb{i}\mathbf{w}_i^\top \mathbf{x}} e^{\mathbb{i}\mathbf{w}_i^\top \mathbf{x}'} \end{aligned}$$

where $p(\mathbf{w})$ is set to be the inverse Fourier transform of K, $\mathbb{i} = \sqrt{-1}$, and $\mathbf{w}_i \in \mathbb{R}^d$ is randomly sampled from a data-independent distribution $p(\mathbf{w})$ [16]. Note that we can define a D-dimensional *randomized* feature map $\mathbf{z}(\mathbf{x}) : \mathbb{R}^d \to \mathbb{C}^D$, which can be explicitly constructed as $\mathbf{z}(\mathbf{x}) := [\exp(\mathbb{i}\mathbf{w}_1^\top \mathbf{x}), \ldots, \exp(\mathbb{i}\mathbf{w}_D^\top \mathbf{x})]^\top$. Other definitions are possible: one could for instance expand the exponentials in pairs $[\cos(\mathbf{w}_i^\top \mathbf{x}), \sin(\mathbf{w}_i^\top \mathbf{x})]$, but this increases the mapped data dimensionality to $\mathbb{R}^{2D}$, while approximating exponentials by $[\cos(\mathbf{w}_i^\top \mathbf{x} + b_i)]$, where $b_i \sim \mathcal{U}(0, 2\pi)$, is more efficient (still mapping to $\mathbb{R}^D$) but has proved less accurate [17]. In matrix notation, given n data points, the kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ can be approximated with the explicitly mapped data, $\mathbf{Z} = [\mathbf{z}_1 \cdots \mathbf{z}_n]^\top \in \mathbb{R}^{n \times D}$, and will be denoted as $\hat{\mathbf{K}} \approx \mathbf{Z}\mathbf{Z}^\top$. This property can be used to approximate any shift-invariant kernel. For instance, the RBF kernel can be approximated using $\mathbf{w}_i \sim \mathcal{N}(\mathbf{0}, \sigma^{-2}\mathbf{I})$, $1 \leq i \leq D$. It is also important to notice that the approximation of K with random Fourier features converges in ℓ_2-norm error with $\mathcal{O}(D^{-1/2})$ when using an appropriate random parameter sampling distribution [18].

For the case of kSNR, we have to sample twice, hence obtain two sets of vectors $\mathbf{w}^x$ and $\mathbf{w}^n$ and the associated randomized data and noise matrices $\mathbf{Z}_x$ and $\mathbf{Z}_n$. On the one hand, a Randomized kSNR for feature extraction trivially reduces to solve the SNR transformation using the explicitly mapped data in the randomized feature space, which is equivalent to solve the generalized eigenproblem $\mathbf{Z}_x^\top \mathbf{Z}_x \mathbf{v}_i = \lambda_i \mathbf{Z}_n^\top \mathbf{Z}_n \mathbf{v}_i$, where we can actually extract a maximum of D features, $D \ll n$. On the other hand, a Randomized kSNR for kernel least squares regression reduces to solve $\boldsymbol{\alpha} = (\mathbf{Z}_x^\top \mathbf{Z}_x + \lambda \mathbf{Z}_n^\top \mathbf{Z}_n)^{-1} \mathbf{Z}_x^\top \mathbf{y}$, where the (now explicit) noise covariance matrix in the randomized feature space acts again as a regularizer. The associated cost by using the random features approximation now reduces to $\mathcal{O}(nD^2)$. Figure 2(b) shows the computational cost as a function of D for a toy example.

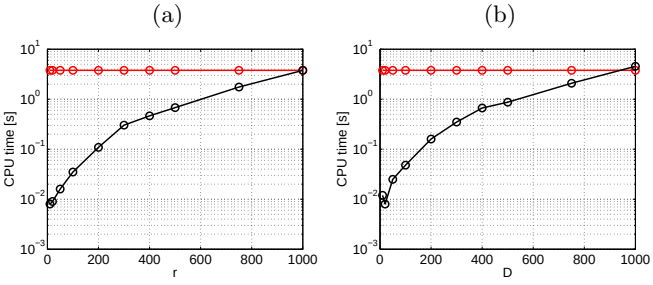

Figure 2: Average computational cost, CPU time [s], over 10 realizations as a function of r and D for the (a) reduced rank and (b) randomized kSNR in black lines (red lines denote full kSNR solution). We used a synthetic example of $n = 1000$ samples drawn from a sigmoid in a 10-dimensional space buried in i.i.d. noise, $\mathcal{N}(0, 0.2)$, and varied r and D accordingly.

3.3 Stability of the kSNR

The use of kSNR in practice raises the question of the convergence of the algorithm with the amount of data available and how the performance changes depending on the dataset at hand. Such results have been previously derived for the particular case of kPCA [19], and can be used to analyze the kSNR properties. Actually, defining $\mathbf{K}^* = (\mathbf{K}_{xn}\mathbf{K}_{nx})^{-1}\mathbf{K}^2$, Theorems 1 and 2 in [19] apply to the kSNR, and provide the upper bounds for the largest and smallest eigenvalues. Depending on how much non-diagonal is $\mathbf{K}^*$, i.e. how large the signal-to-noise relations are, the bounds may be tighter than those of kPCA. With an appropriate estimation of the noise structure, and tuning of the kernel parameters, the performance of kSNR will be at least as fitted as that of kPCA.

4 Experimental results

This section presents the results of different kSNR methods in several signal processing and machine learning problems.

Typical kernel functions are the linear $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$, the polynomial $K(\mathbf{x}_i, \mathbf{x}_j) = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + 1)^d$, $d \in \mathbb{Z}^+$, and the Radial Basis Function (RBF), $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2\right)$, $\sigma \in \mathbb{R}^+$. In the experiments, the RBF kernel function is used[1]. Therefore, only two parameters have to be tuned: the kernel width (σ) and the regularization constant (λ). In order to select the optimal parameters, we split the data into two sets of equal size for validation purposes. Values tested for kernel width are obtained as the mean distance between training points multiplied by a factor in the range $[10^{-3}, 10^3]$; and values for the regularization constant λ are tested in the range $[10^{-3}, 10^3]$. After following this grid-search approach, the best parameters values in terms of accuracy are selected by cross-validation.

[1]Note that specific applications might need particular kernel functions.

4.1 Experiment 1: Function approximation

First we showcase the behavior of the kSNR as a regression technique under non-Gaussian i.i.d. noise. For this synthetic experiment, we generate 1000 data points from a sinc function $s_t = \sin(t)/t$, with $t \in [-\pi, +\pi]$, with the addition of a variety of noises, $y_t = s_t + n_t$: 1) Gaussian, $n_t \sim \mathcal{N}(0, \sigma_n^2)$; 2) Uniform, $n_t \sim \mathcal{U}(0, 1)$; 3) Poisson, $n_t \sim \mathcal{P}(\lambda)$, $\lambda \in [0, 0.3]$; 4) Scale-dependent multiplicative, $n_t = m_t \times |s_t|$ where $m_t \sim \mathcal{N}(0, \sigma_n^2)$. In order to assess the performance, we partition the data into two sets of equal size, for cross-validation and testing respectively.

The comparison of KRR and kSNR is presented in Fig. 3. We also show a baseline in blue, which represents the SNR of the original noise-free data points s_n and is effectively a lower bound on the performance. This simple –yet informative– toy problem, clearly motivates the relevance of this work: while in the first two cases (Gaussian and uniform) both approaches work similarly, the differences become clear in the later two (Poisson and scale dependent). This suggests that we can exploit the nice properties of kSNR in scenarios that involve non-Gaussianity or correlated noise.

4.2 Experiment 2: Channel equalization

This experiment consists in equalizing a binary pulse amplitude modulation signal at the output of a dispersive channel, whose low-pass model was a tapped delay line with $h = \delta_i + 0.6\delta_{i-1} + 0.2\delta_{i-2} - 0.1\delta_{i-3} + 0.01\delta_{i-4}$. This impulse response can represent a minimum-phase dispersive channel, which is common in suburban and hilly terrain environments. We synthesized $N = 128$ random binary values $y_i \in \{0, 1\}, i = 1, \ldots, N$, that are transmitted through the previous channel h, and eventually corrupted by an additive noise n. Therefore, the received signal at the end is $x = h * y + n$, from which we try to estimate the transmitted signal y. Half of the samples were allocated to train a LS-SVM classifier and the remaining samples were used for validation purposes to select the optimal parameters. As an assessment of the per-

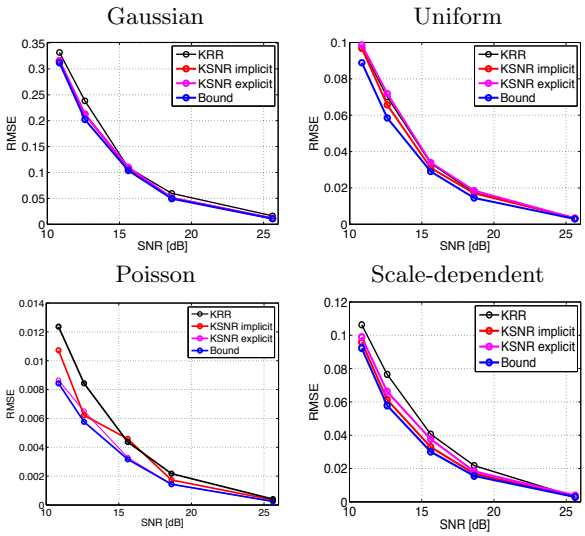

Figure 3: Regression experiment: estimation of a sinc. RMSE vs SNR for four types of noise.

formance, we studied the bit error rate (BER) of an independent burst of test 10^5 samples under additive and scale-dependent noise n drawn from a gamma distribution. We show the average results after 10 random iterations, for each SNR in the range of +6 to +20 dB. In Fig. 4 we compare the performance of the different classifiers. The explicit version of kSNR provides the best results, especially for SNR values under 12dB (improvements in the order of 4dB).

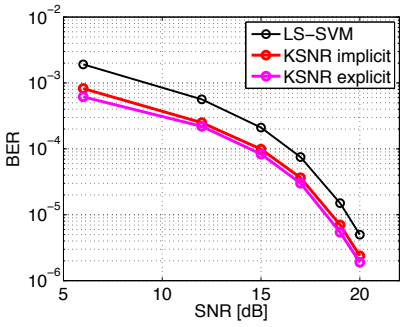

Figure 4: Channel equalization experiment: Bit error rate (BER) vs SNR.

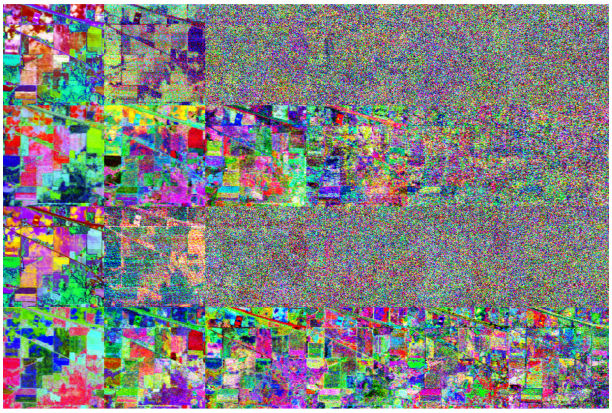

Figure 5: First 18 features extracted from AVIRIS bands. From top to bottom: PCA, MNF, KPCA, and explicit kSNR in the kernel space. From left to right: subimages (RGB composites) with triplets of extracted principal components in descending order of relevance.

4.3 Experiment 3: Hyperspectral image feature extraction

This experiment illustrates the method's capabilities in a challenging feature extraction and subsequent classification problem. In particular, we first reduce the dimensionality of a hyperspectral image acquired by the airborne AVIRIS sensor[2] and then use the extracted features for classification. The image consists of 145×145 pixels, and 10366 of them are labeled into 16 agricultural classes (ground truth). Each pixel contains 220 contiguous spectral bands, including 20 channels in the spectral region affected by atmospheric water vapor absorptions, which present high noise levels [20]. Therefore, we reduce the data dimensionality by extracting features from the original 220 channels and benchmark the kSNR performance against standard PCA, MNF (aka SNR), and KPCA. The quality of the first 18 extracted principal components is analyzed in Fig. 5 by sorting them from higher to lower importance (eigenvalues). Visual inspection reveals that kSNR provides the most noise-free image features.

[2] https://engineering.purdue.edu/~biehl/

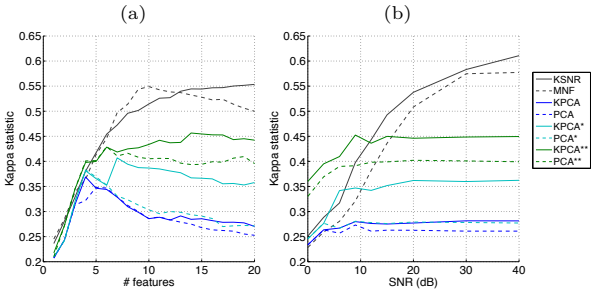

Figure 6: LDA classification accuracy (kappa statistics, κ) as a function of: (a) the number of extracted features used as inputs in LDA; (b) the SNR for different levels of additive noise. Three scenarios are considered for the PCA based methods: 1) using noisy data for both finding the transform (training) and extracting the features (testing); 2) using denoised samples to train the transform and then extract the features from noisy data (*); and 3) denoising both the train and test datasets (**).

In order to test the method's performance we added different levels of Gaussian noise to the original image (SNR from 0 to 40 dB) and then used the extracted features as input for a linear discriminant analysis (LDA) classifier. Figure 6(a) shows the classification accuracy as a function of the number of extracted features for a SNR of 20 dB. The proposed kSNR and MNF provide the best accuracy when confronted with the linear and kernel PCA versions, which stresses the importance of accounting for the noise contribution. When the data is denoised before computing the PCA/KPCA transform (**), the results are also better but lower than for the proposed method, which illustrates that characterizing the noise distribution and avoiding directions affected by noise might be more robust than estimating the noise and then subtracting it from each independent sample. Figure 6(b) shows the classification accuracy for different levels of additive Gaussian noise when extracting 15 principal components. Under extreme noise conditions (SNR=0dB) the noise characteristics (e.g. noise covariance) are poorly estimated and thus the proposed method shows low accuracy. However, working in less than 10dB is far from being realistic in most applications, and all algorithms perform poorly in those regimes. When the SNR increases, the explicit kSNR method outperforms the other approaches. It is also worth noting that kernel methods provide better results than their linear counterparts in all cases. Finally, visual inspection of the classification maps for a SNR of 20 dB using the best sets of extracted features reveals that kSNR provides more uniform land cover maps (Fig. 7).

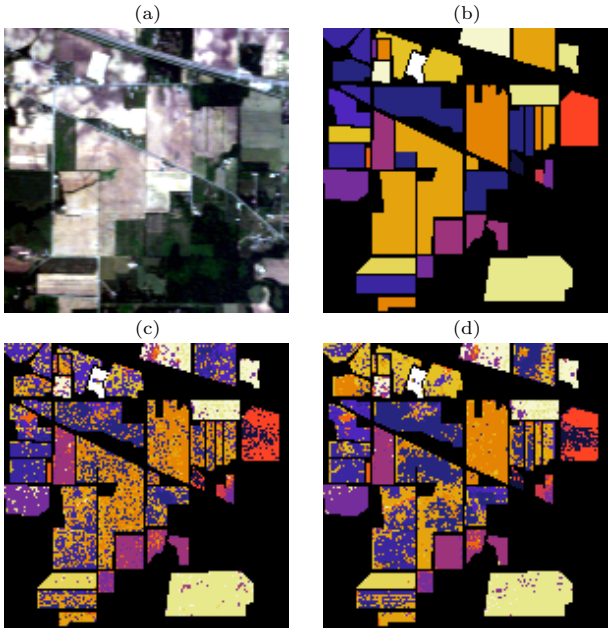

Figure 7: (a) AVIRIS scene presenting vegetated crops and bright bare soils; (b) Ground Truth of the 16 land-cover classes; and (c) MNF and (d) explicit kSNR classification maps.

4.4 Experiment 4: Causal discovery

Establishing causal relations among random variables using empirical data is perhaps the most important challenge in today's Science. In this experiment, we use kSNR for causal discovery in bivariate simultaneous data. To this end, following [21] we aim at inferring causal links between two observed random variables x and y. The experiment is designed to first

predict y from x. Then we measure the independence of the residual $r_f = y - f(x)$ and the independent (potential cause) variable x. Equivalently, we follow the same procedure with the backward estimation x from y, and check for independence of $r_b = x - g(y)$ from y. The direction leading to most independent residuals suggests the causing mechanism. In the standard approach [21], Gaussian processes were used for fitting. Noting that noise additivity can break in real scenarios, we aim here to compare the detection accuracy and sensitivity using both the implicit and explicit kSNR to compensate for non-additive noise.

For illustration purposes, we use a standard example where the data comes from 349 german weather stations[3] that collected both altitude (meters) and average temperature per year (^{o}C). We split the data evenly into cross-validation and test sets. We measure how independent these variables are using the p-values from HSIC [22, 23]. In order to get a reliable estimate of the noise, this was computed as the difference between the k-nearest neighbors least squares approximation minus the observed signal [14]. Table 1 confirms that the different approaches correctly infers a causal link from 'altitude' to 'temperature', however, interestingly, the p-values corresponding to (explicit) kSNR are significantly smaller, and the difference between p_f and p_b becomes smaller, yet more realistic.

Table 1: 'Altitude (x) causes temperature (y)'

Method	p_f	p_b	Conclusion
KRR	2.88×10^{-2}	3.54×10^{-12}	$x \to y$
Implicit kSNR	7.47×10^{-4}	9.28×10^{-11}	$x \to y$
Explicit kSNR	2.94×10^{-16}	8.83×10^{-23}	$x \to y$

5 Conclusions

This paper presented the kernel signal-to-noise ratio for some of the most relevant tasks in machine learning, namely, feature extraction, regression, classification, and causal discovery. This approach provides a regularizer that successfully deals with non-linear signal-to-noise relations. Two alternative formulations have been presented to reduce the computational cost for large-scale problems and the stability of the method has been analyzed. The empirical evaluation shows that the kSNR compares favorably with the corresponding state-of-the-art methods for each of these problems, particularly when dealing with correlated or non-Gaussian noise. Additionally, both implicit and explicit estimation of the noise were discussed and evaluated. Interestingly, the explicit formulation typically turns out to be more accurate and requires a lower computational burden. Future work will deal with the design of accurate noise estimation techniques in rkHs, the extension of the approaches to estimate conditional independence, and further evaluation in challenging causal discovery problems.

[3] http://webdav.tuebingen.mpg.de/cause-effect/

Acknowledgments

This work has been supported by the Spanish Ministry of Economy and Competitiveness (MINECO) under projects TIN2015-64210-R and TEC2016-77741-R (ERDF), and by the ERC Consolidator Grant SEDAL ERC-2014-CoG 647423.

The authors would like to thank Dr. Allan A. Nielsen at the Danmarks Tekniske Universitet (DTU), and Dr. Robert Jenssen at the University of Trømso (UiT) for useful comments on this work.

References

[1] A. A. Green, M. Berman, P. Switzer, and M. D. Craig, "A transformation for ordering multispectral data in terms of image quality with implications for noise removal," *IEEE Trans. Geosc. Rem. Sens.*, vol. 26, no. 1, pp. 65–74, 1998.

[2] A. A. Nielsen, "Kernel maximum autocorrelation factor and minimum noise fraction transformations," *IEEE Trans. Image Processing*, vol. 20, pp. 612–624, Mar. 2011.

[3] L. Gómez-Chova, A. A. Nielsen, and G. Camps-Valls, "Explicit signal to noise ratio in reproducing kernel Hilbert spaces," in *IGARSS*, pp. 3570–3573, Jul 2011.

[4] M. J. Canty and A. A. Nielsen, "Linear and kernel methods for multivariate change detection," *Computers & Geosciences*, vol. 38, no. 1, pp. 107–114, 2012.

[5] A. Christiansen, J. Carstensen, F. Møller, and A. Nielsen, "Monitoring the change in colour of meat: A comparison of traditional and kernel-based orthogonal transformations," *Journal of Spectral Imaging*, vol. 3, no. 1, p. a1, 2012.

[6] L. Gómez-Chova and G. Camps-Valls, "Learning with the kernel signal to noise ratio," in *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, pp. 1–6, Sep 2012.

[7] Y. Jia, F. Nie, and C. Zhang, "Trace ratio problem revisited," *IEEE Transactions on Neural Networks*, vol. 20, no. 4, pp. 729–735, 2009.

[8] B. Schölkopf and A. Smola, *Learning with Kernels – Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.

[9] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[10] F. Riesz and B. S. Nagy, *Functional Analysis*. Frederick Ungar Publishing, 1955.

[11] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, eds., *Least Squares Support Vector Machines,*. Singapore: World Scientific Pub. Co., 2002.

[12] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. Smola, and K.-R. Müller, "Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces," *IEEE Trans. Patt. Anal. Mach. Intell*, vol. 25, pp. 623–628, 2003.

[13] M. L. Braun, J. M. Buhmann, and K.-R. Müller, "On relevant dimensions in kernel feature spaces," *Journal of Machine Learning Research*, vol. 9, pp. 1875–1908, 2008.

[14] H. Pi and C. Peterson, "Finding the embedding dimension and variable dependencies in time series.," *Neural Computation*, vol. 6, no. 3, pp. 509–520, 1994.

[15] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, (USA), pp. 1177–1184, Curran Associates Inc., 2007.

[16] A. Rahimi and B. Recht, "Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning," in *Advances in Neural Information Processing Systems 21*, pp. 1313–1320, Curran Associates, Inc., 2009.

[17] J. Sutherland and J. Schneider, "On the error of random fourier features," in *UAI*, pp. 862–871, 2015.

[18] L. K. Jones, "Annals of statistics," *A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training*, vol. 20, pp. 608–613, 1992.

[19] J. Shawe-Taylor, C. K. I. 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*, vol. 51, no. 7, pp. 2510–2522, 2005.

[20] J. Arenas-García, K. B. Petersen, G. Camps-Valls, and L. K. Hansen, "Kernel multivariate analysis framework for supervised subspace learning: A tutorial on linear and kernel multivariate methods," *IEEE Sig. Proc. Mag.*, vol. 30, no. 4, pp. 16–29, 2013.

[21] P. O. Hoyer, D. Janzing, J. M. Mooij, J. Peters, and B. Schölkopf, "Nonlinear causal discovery with additive noise models," in *NIPS*, pp. 689–696, 2008.

[22] A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. Smola, "A kernel statistical test of independence," in *Advances in Neural Information Processing Systems 20*, pp. 585–592, MA: MIT Press, 2008.

[23] G. Camps-Valls, J. Mooij, and B. Schölkopf, "Remote sensing feature selection by kernel dependence measures," *IEEE Geosc. Rem. Sens. Lett.*, vol. 7, pp. 587–591, Jul 2010.