Learning via variably scaled kernels

Learning via variably scaled kernels

C. Campi*, F. Marchetti+, E. Perracchione°
+ Dipartimento di Matematica "Tullio Levi-Civita",
Università di Padova,
* Dipartimento di Salute della Donna e del Bambino,
Università di Padova,
° Dipartimento di Matematica DIMA, Università di Genova,
cristina.campi@unipd.it
francesco.marchetti.1@phd.unipd.it
perracchione@dima.unige.it

Abstract

We investigate the use of the so-called Variably Scaled Kernels (VSKs) for learning tasks, with a particular focus on Support Vector Machine (SVM) classifiers and Kernel Regression Networks (KRNs). Concerning the kernels used to train the models, under appropriate assumptions, the VSKs turn out to be *more expressive* and *more stable* than the standard ones. Numerical experiments and applications to breast cancer and COronaVIrus Disease 19 (COVID19) data support our claims. For the practical implementation of the VSK setting, we need to select a suitable *scaling function*. To this aim, we propose different choices, including for SVMs a probabilistic approach based on the Naive Bayes (NB) classifier. For the classification task, we also numerically show that the VSKs can be seen as an alternative to the sometimes computationally demanding feature extraction procedures.

1 Introduction

In the context of approximation theory, the Variably Scaled Kernels (VSKs) were introduced in 2015 by [6]. The basic idea behind them is to map the initial set of examples via a scaling function and construct an augmented approximation space. In this sense, they can be seen as a generalisation of *feature augmentation strategies*. Indeed, all methods based on feature augmentation, as e.g. zero padding and feature replication [9, 21, 26], fall into the general VSK setting that we are going to investigate.

Focusing on kernel learning methods and specifically on KRNs and SVMs (see e.g. [16, 41]), we give a very general formulation of feature augmentation schemes via VSKs. In doing so, we drive our attention towards the Gaussian

and linear kernels, being truly popular for learning issues. We provide theoretical results concerning their practical implementation, expressiveness [13] and we further analyze the spectrum of the kernel matrices constructed via VSKs. This study reveals the effectiveness of the proposed approach especially for the Gaussian kernel, indeed the condition number of the VSK kernel matrix is less than or equal to the condition number of the matrix constructed via the standard kernel. This fact turns out to be meaningful for KRNs, where one may require to compute the inverse of the kernel matrix, which is usually affected by severe ill-conditioning. Moreover, for the selection of the scaling function of the KRN-VSK, one can refer to the available literature in the context of approximation theory [10, 35]. Indeed, the scaling function might be selected so that it *mimics* the samples and this might lead to an improvement in terms of accuracy and/or stability (see e.g. [6, 10, 11]). Here in particular we propose to use a non-linear fitting of the function itself as augmented feature.

While for the KRN-VSK we can refer to some available literature for selecting the scaling function, for SVM-VSK we need to take into account also probabilistic solutions. More precisely, focusing on binary classification problems, we first note that the VSK setting induces new feature maps and spaces that depend on the scaling function associated to the VSK. For being competitive with the accuracy of the classical SVMs, as well as with other common classifiers, we have to select a *suitable* scaling function for the VSKs. To this aim, we remark that the SVM is characterized by a *geometric* point of view. Nevertheless, methods based on probability distributions, as the NB classifiers, might outperform SVM. For that reason, many efforts are devoted to investigate which classifier performs better and under which conditions; for a general overview refer e.g. to [7, 30, 46]. In this work we thus fuse SVM and NB classifiers by means of VSKs, so that the mixed approach takes into account the probabilistic features of the NB algorithm and classifies geometrically with SVM. In view of this, our method can also be *substituted* to feature extraction schemes; refer e.g. to [19, 44]. Indeed, we numerically show that we are able to randomly reduce the dataset and then encode the missing information via SVM-VSK.

The paper is organized as follows. In Section 2, we briefly review the use of kernels in machine learning literature. In Section 3, we investigate the VSKs for two learning methods, specifically SVM and KRNs. Then, in Sections 4 and 5, we drive our attention towards the Gaussian and linear VSKs as well as towards the problem of selecting the scaling function. Section 6 is devoted to numerical experiments with both toy models and real datasets. The last section deals with conclusions and work in progress.

2 Preliminaries

We consider a learning problem with training examples

$$\Sigma = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\},$$

where $\boldsymbol{x}_i \in \Omega \subseteq \mathbb{R}^n$ and $y_i \in \mathbb{R}$. For the particular case of the classification setting, we fix $y_i \in \{-1, +1\}$.

For both SVMs and KRNs, we drive our attention towards (strictly) positive definite kernels $\kappa : \Omega \times \Omega \longrightarrow \mathbb{R}$, where Ω is a bounded set, that can be decomposed via the Mercer's Theorem as explained below (see e.g. Theorem 2.2. [15] p. 107 or [25]).

Theorem 2.1 *Let $\kappa : \Omega \times \Omega \longrightarrow \mathbb{R}$ be a continuous (strictly) positive definite kernel that satisfies*

$$\int_\Omega \kappa(\boldsymbol{x}, \boldsymbol{y}) v(\boldsymbol{x}) v(\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \geq 0, \quad \forall v \in L_2(\Omega), \boldsymbol{x}, \boldsymbol{y} \in \Omega,$$

then the kernel can be expressed as

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k \geq 0} \lambda_k \rho_k(\boldsymbol{x}) \rho_k(\boldsymbol{y}), \quad \boldsymbol{x}, \boldsymbol{y} \in \Omega,$$

where $\{\lambda_k\}_{k \geq 0}$ are the (non-negative) eigenvalues and $\{\rho_k\}_{k \geq 0}$ are the (L_2-orthonormal) eigenfunctions of the operator $T : L_2(\Omega) \longrightarrow L_2(\Omega)$, given by

$$T[v](\boldsymbol{x}) = \int_\Omega \kappa(\boldsymbol{x}, \boldsymbol{y}) v(\boldsymbol{y}) d\boldsymbol{y}.$$

Moreover, such expansion is absolutely and uniformly convergent.

For such kernels that admit a Mercer expansion (also called valid kernels according to the definition given by [41]), it is worth to note that we can interpret the series representation in terms of an inner product in the so-called *feature space* F, which is a Hilbert space. Indeed,

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) \rangle_F, \quad \boldsymbol{x}, \boldsymbol{y} \in \Omega, \tag{2.1}$$

where $\Phi : \Omega \longrightarrow F$ is a *feature map*. For a given kernel, the feature map and space are not unique. A possible solution is the one of taking the map $\Phi(\boldsymbol{x}) = \kappa(\cdot, \boldsymbol{x})$, which is linked to the characterization of F as a reproducing kernel Hilbert space; see [16, 41] for further details.

In the classification context, many studies are devoted to investigate and measure the *complexity* of a chosen model, such as the so-called VC dimension [43] and the empirical Rademacher complexity [4]. The complexity of a method is usually referred to as *capacity* or *expressiveness*. Indeed, complex models have the capability to perform complex tasks, by determining elaborated decision functions, and thus to express sophisticated links between the data. In any case, the capacity of a method needs to be tailored to the considered task, in order to avoid overfitting; for a general overview, we refer e.g. the reader to [38].

To better investigate the concept of expressiveness in the kernel setting, we introduce the kernel matrix K constructed via the dataset $\Xi = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subseteq \Omega$, i.e. the matrix of entries

$$\mathsf{K}_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad i, j = 1, \ldots, N, \tag{2.2}$$

where κ is a (strictly) positive definite kernel. Note that if κ is a strictly positive definite kernel then K is positive definite, while it is positive semi-definite if κ is a positive definite kernel.

Remark 2.1 *The expressiveness of a kernel-based model is related to the number of dichotomies achievable by a linear separator in the feature space. Moreover, concerning the rank of the kernel matrix, we have the following result [13, Theorem 2, p. 7].*

Theorem 2.2 *Let* K *be the kernel matrix as in* (2.2) *constructed via* $\Xi = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subseteq \Omega$*, let us denote by* $\operatorname{rank}(\mathsf{K})$ *its rank. Then, there exists at least one subset of examples of size* $\operatorname{rank}(\mathsf{K})$ *that can be shattered by a linear function.*

As capacity measure dedicated to the kernel setting, we consider the *spectral ratio* that has been introduced in [13]. It is defined as

$$S(\mathsf{K}) = \frac{\operatorname{tr}(\mathsf{K})}{\|\mathsf{K}\|_{\mathsf{F}}} = \frac{\sum_{i=1}^{N} \mathsf{K}_{ii}}{\sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} \mathsf{K}_{ij}^2}}.$$

According to the following definition, see [13, Definition 1, p. 8], such quantity is an expressiveness measure for kernels. As a remark, we also point out that it is connected to the empirical Rademacher complexity [13, Theorem 4, p. 9].

Definition 2.1 *Let* $\kappa_i, \kappa_j : \Omega \times \Omega \longrightarrow \mathbb{R}$*, be two (strictly) positive definite kernels. We say that* κ_j *is more specific (or more expressive) than* κ_i *whenever for any dataset* $\Xi = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subseteq \Omega$*, we have*

$$S(\mathsf{K}^i) \leq S(\mathsf{K}^j),$$

where K^i *and* K^j *are the kernel matrices on* Ξ *obtained via* κ_i *and* κ_j*, respectively.*

Being the spectral ratio an expressiveness measure, it is related to the rank of the kernel matrix (see also Remark 2.1), indeed

$$1 \leq S(\mathsf{K}) \leq \sqrt{\operatorname{rank}(\mathsf{K})}.$$

We conclude this brief review on kernels for machine learning by pointing out that the kernel matrices introduced above might suffer from severe ill-conditioning. In order to partially overcome instability issues in the approximation framework, a possible solution comes from the use of VSKs (see below for their definition), which have been recently introduced in [6]; refer also to [10, 11].

Definition 2.2 *Let $\Lambda \subseteq \mathbb{R}$ be a bounded set. Let $\kappa : \tilde{\Omega} \times \tilde{\Omega} \longrightarrow \mathbb{R}$, $\tilde{\Omega} = \Omega \times \Lambda \subseteq \mathbb{R}^{n+1}$, be a continuous (strictly) positive definite kernel. Given a scaling function $\psi : \Omega \longrightarrow \Lambda$, a variably scaled kernel $\kappa^{\Psi} : \Omega \times \Omega \longrightarrow \mathbb{R}$ is defined as*

$$\kappa^{\Psi}(\boldsymbol{x}, \boldsymbol{y}) := \kappa((\boldsymbol{x}, \psi(\boldsymbol{x})), (\boldsymbol{y}, \psi(\boldsymbol{y})),$$

for $\boldsymbol{x}, \boldsymbol{y} \in \Omega$.

When dealing with Mercer's kernels, the construction of a VSK as in Definition 2.2 provides a valid kernel. We now extend this general setting to work with KRNs and SVMs.

3 Learning with VSKs

To have a clear theoretical framework, we investigate the use of VSKs as a feature augmentation algorithm, where *new* features are added to the original dataset in order to possibly increase the performances of learning schemes.

At first, we give a multidimensional extension of the scaling function. Thus, let $\Lambda \subseteq \mathbb{R}^m$, $m > 0 \in \mathbb{N}$ be a bounded set. Given $\boldsymbol{x} \in \Omega$, let $\psi : \Omega \longrightarrow \Lambda$ be a function that extracts m features from $\boldsymbol{x}$. Then, letting $\tilde{\Omega} = \Omega \times \Lambda$, we can define a function $\Psi : \Omega \longrightarrow \tilde{\Omega}$ as

$$\Psi(\boldsymbol{x}) \coloneqq (\boldsymbol{x}, \psi(\boldsymbol{x})).$$

The function Ψ extends the data vector $\boldsymbol{x} \in \Omega$, including m features that depend on the original ones.

Thus, in the VSK setting, we define the (strictly) positive definite kernel $\kappa^{\Psi} : \Omega \times \Omega \longrightarrow \mathbb{R}$, given by

$$\kappa^{\Psi}(\boldsymbol{x}, \boldsymbol{y}) = \kappa\left(\Psi(\boldsymbol{x}), \Psi(\boldsymbol{y})\right),$$

where $\kappa : \tilde{\Omega} \times \tilde{\Omega} \longrightarrow \mathbb{R}$ is a (strictly) positive definite kernel. The kernel κ^{Ψ} is a valid kernel, namely that it corresponds to an inner product in the associated feature space F_{ψ} (see [41, Proposition 3.22, page 75]). Moreover, it induces a *new* feature map $\Theta : \Omega \longrightarrow F_{\psi}$ so that

$$\kappa^{\Psi}(\boldsymbol{x}, \boldsymbol{y}) = \langle \Theta(\boldsymbol{x}), \Theta(\boldsymbol{y}) \rangle_{F_{\psi}}. \tag{3.1}$$

Referring to equation (2.1), because of [6, Theorem 3.1], the spaces F_{ψ} and the classical feature space F, associated to $\kappa : \tilde{\Omega} \times \tilde{\Omega} \longrightarrow \mathbb{R}$ and induced by the feature map $\Upsilon : \tilde{\Omega} \longrightarrow F$, are isometric; see also [10, Proposition 2.3].

We now investigate the use of the VSKs for both SVMs and KRNs.

3.1 SVM-VSK

In this section, we present the VSK setting in the SVM algorithm. For this general overview, we also refer the reader to [16, 41].

We take $\Xi = \{\boldsymbol{x}_i,\ i = 1, \dots, N\} \subseteq \Omega$, where $\Omega \subseteq \mathbb{R}^n$ is a bounded set. The associate function values are so that $y_i \in \{-1, +1\}$, $i = 1, \dots, N$. Indeed, for the binary classification problem via VSKs we need to find a predictor, i.e. a decision function $s^{\Psi} : \Omega \longrightarrow \{-1, +1\}$, that assigns appropriate labels, i.e. $\tilde{y}_i \in \{-1, +1\}$, to other unknown samples $\tilde{\boldsymbol{x}}_i$, $i = 1, \dots, t$.

Given $\boldsymbol{x} \in \Omega$, we define a non-linear SVM classifier that makes use of VSKs via the following decision function:

$$s^{\Psi}(\boldsymbol{x}) = \operatorname{sign}(h^{\Psi}(\boldsymbol{x})) \coloneqq \operatorname{sign}(\Theta(\boldsymbol{x})\boldsymbol{w}^{\intercal} + b),$$

where $\Theta : \Omega \longrightarrow F_{\psi}$ is the VSK feature map, $\boldsymbol{w} = (w_1, \dots, w_n) \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are given by

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \Theta(\boldsymbol{x}_i),$$

and

$$b = y_i - \sum_{j=1}^{N} \alpha_j \kappa^{\Psi}(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{3.2}$$

The coefficients $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_N) \in \mathbb{R}^N$ are the solution of the following *soft margin* problem [16]

$$\begin{cases} \min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \dfrac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \kappa^{\Psi}(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{i=1}^{N} \alpha_i, \\ \text{s.t. } \sum_{i=1}^{N} \alpha_i y_i = 0, \\ 0 \leq \alpha_i \leq \zeta, \quad i = 1, \dots, N, \end{cases}$$

where $\zeta \in \mathbb{R}_+ = [0, +\infty)$ in known as *bounding box*. The equation of the SVM decision function $s^{\Psi} : \Omega \longrightarrow \{-1, +1\}$, i.e. $\boldsymbol{w}$ and b as in equation (3.1) and (3.2), is then found by imposing the Karush Kuhn Tucker conditions (see e.g. [28]) and thanks to (3.1), for $\boldsymbol{x} \in \Omega$, it reads as follows

$$s^{\Psi}(\boldsymbol{x}) = \operatorname{sign}(h^{\Psi}(\boldsymbol{x})) = \operatorname{sign}(\Theta(\boldsymbol{x})\boldsymbol{w}^{\intercal} + b) = \operatorname{sign}\left(\sum_{i=1}^{N} y_i \alpha_i \kappa^{\Psi}(\boldsymbol{x}, \boldsymbol{x}_i) + b \right).$$

Note that in equation (3.2) i denotes the index of an α_i so that $0 < \alpha_i < \zeta$. For stability purposes, b is usually evaluated as an average among all candidates. If one uses the standard kernel $\kappa : \Omega \times \Omega \longrightarrow \mathbb{R}$, then we recover the classical SVM setting.

As a second test case for the use of VSKs in machine learning context, we investigate regression networks.

3.2 KRN-VSK

For the purposes of KRN, given $\Xi = \{\boldsymbol{x}_i,\ i = 1, \dots, N\} \subseteq \Omega$, where $\Omega \subseteq \mathbb{R}^n$ is a bounded set, we fix the output variables $y_i \in \mathbb{R}$, $i = 1, \dots, N$. Indeed, KRNs

are used for regression/interpolation purposes, hence, e.g. they turn out to be meaningful for studying the behaviour of longitudinal data.

Concerning supervised learning networks, the simplest strategy consists in learning the trend between inputs and outputs via a predictor $s^{\Psi} : \Omega \longrightarrow \mathbb{R}$ which is a linear combination of *some* basis functions, in this case VSKs. For a general overview on KRNs, we refer the reader to [16, 29].

We keep the general framework of KRNs and we adapt them to the use of VSKs. Here, we focus on kernels with centers at locations $Z = \{\boldsymbol{z}_i,\ i = 1, \ldots, M\} \subseteq \Omega$, and thus our KRN-VSK predictor $s^{\Psi} : \Omega \longrightarrow \mathbb{R}$ is of the form

$$s^{\Psi}(\boldsymbol{x}) = \sum_{i=1}^{M} c_i \kappa^{\Psi}(\boldsymbol{x}, \boldsymbol{z}_i), \tag{3.3}$$

for (strictly) positive definite kernels $\kappa^{\Psi} : \Omega \times \Omega \longrightarrow \mathbb{R}$ and for some real coefficients $c_1, \ldots, c_M$. The learning function (3.3) is thus the simplest neural network which involves only one linear combination of basis functions, i.e. one layer.

For KRN-VSK, we compute $\boldsymbol{c} = (c_1, \ldots, c_M) \in \mathbb{R}^M$ via the following minimization problem [15]

$$\min_{\boldsymbol{c} \in \mathbb{R}^M} \left[\sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{M} c_j \kappa^{\Psi}(\boldsymbol{x}_i, \boldsymbol{z}_j) \right)^2 + \nu \sum_{j=1}^{M} c_j^2 \right],$$

where $\nu \in \mathbb{R}_+$ is a regularization parameter.

In the following we may take the set of kernel centres $Z \equiv \Xi$. In that case the kernel matrix K^{Ψ} of entries

$$\mathsf{K}^{\Psi}_{ij} = \kappa^{\Psi}(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad i, j = 1, \ldots, N,$$

is square. Furthermore, if a strictly positive definite kernel as the Gaussian function is used, then the matrix is non-singular. Therefore, we may look to the special setting for which $\nu = 0$. In that case, the solution can be found as $\boldsymbol{c}^{\intercal} = (\mathsf{K}^{\Psi})^{-1} \boldsymbol{y}^{\intercal}$, where $\boldsymbol{y} = (y_1, \ldots, y_N)$ and $\boldsymbol{c} = (c_1, \ldots, c_N)$.

In general, computing the inverse of the kernel matrix K might lead to serious instability issues due to the typical ill-conditioning of the kernel matrix. This problem may be somehow overcome by selecting a *safe shape parameter* γ, formally introduced below, and/or by using stable bases; refer e.g. to [20, 33]. Moreover, we will point out both numerically and theoretically in the incoming sections that a performing alternative to reduce the ill-conditioning comes from the use of VSKs.

4 Gaussian and linear VSKs

In this section, we focus on specific kernels providing the practical implementation of the variably scaled setting. Furthermore, we also study the expressiveness and the conditioning induced by the VSKs.

4.1 Gaussian kernel

Radial kernels are truly common. They are kernels for whom there exists a Radial Basis Function (RBF) $\varphi : [0, \infty) \longrightarrow \mathbb{R}$ and (possibly) a shape parameter $\gamma > 0$ such that, for all $\boldsymbol{x}, \boldsymbol{y} \in \Omega$,

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \kappa_\gamma(\boldsymbol{x}, \boldsymbol{y}) = \varphi_\gamma(||\boldsymbol{x} - \boldsymbol{y}||_2) := \varphi(r).$$

Among all radial kernels, we remark that the Gaussian is given by

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \kappa_\gamma(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{e}^{-\gamma \|\boldsymbol{x}-\boldsymbol{y}\|_2^2} = \mathrm{e}^{-\gamma r^2} := \varphi(r).$$

We now discuss its practical implementation in the variably scaled setting. We point out that the Gaussian kernel is strictly positive definite and thus its associated kernel matrix turns out to be positive definite; see e.g. [16].

Practical implementation for the Gaussian VSK

Throughout this section we take N distinct data $\Xi = \{\boldsymbol{x}_i,\ i = 1, \ldots, N\} \subseteq \Omega$, where $\Omega \subseteq \mathbb{R}^n$ is a bounded set. Moreover, let $\Lambda \subseteq \mathbb{R}^m$ be a bounded set.

The Gaussian VSK matrix can be seen as a Hadamard product, indeed we have the following result.

Theorem 4.1 *Let $\Xi = \{\boldsymbol{x}_i,\ i = 1, \ldots, N\} \subseteq \Omega$ be a set of distinct data. Let $\psi : \Omega \longrightarrow \Lambda$ be the scaling function for the VSK setting. Let $\kappa : \Omega \times \Omega \longrightarrow \mathbb{R}$ be the Gaussian kernel. Then, the VSK matrix constructed on Ξ via $\kappa^\Psi : \Omega \times \Omega \longrightarrow \mathbb{R}$ is given by*

$$\mathsf{K}^\Psi = \mathsf{K} \circ \mathsf{K}^\psi,$$

where $\mathsf{K}^\psi_{ij} = \mathrm{e}^{-\|\psi(\boldsymbol{x}_i)-\psi(\boldsymbol{x}_j)\|_2^2}$, $i, j = 1, \ldots, N$, and $\circ$ denotes the Hadamard matrix product.

Proof: For $\boldsymbol{x}, \boldsymbol{y} \in \Omega$, we have that

$$\kappa^\Psi(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{e}^{-\left(\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \|\psi(\boldsymbol{x})-\psi(\boldsymbol{y})\|_2^2\right)} = \mathrm{e}^{-\|\boldsymbol{x}-\boldsymbol{y}\|_2^2}\ \mathrm{e}^{-\|\psi(\boldsymbol{x})-\psi(\boldsymbol{y})\|_2^2}.$$

Therefore, the entries of the VSK matrix built on $\Xi = \{\boldsymbol{x}_i,\ i = 1, \ldots, N\}$ are given by

$$\mathsf{K}^\Psi_{ij} = \mathrm{e}^{-\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2}\ \mathrm{e}^{-\|\psi(\boldsymbol{x}_i)-\psi(\boldsymbol{x}_j)\|_2^2}, \quad i, j = 1, \ldots, N,$$

and thus

$$\mathsf{K}^\Psi = \mathsf{K} \circ \mathsf{K}^\psi.$$

□

About the Hadamard product, we report here a result that can be traced back to 1911 by Schur [39]. It will be helpful in what follows; refer also to [14, Lemma A.5] and [18, Lemma 2.1].

Theorem 4.2 *If* E *and* $\mathsf{M} \in \mathbb{R}^{N\times N}$ *are positive definite matrices, denoting by* $\lambda_{\min}$ *and* $\lambda_{\max}$ *the smallest and largest eigenvalue of a matrix, we have that*

$$\lambda_{\min}(\mathsf{E}) \min_{i=1,\dots,N} \mathsf{M}_{ii} \leq \lambda_i(\mathsf{E}\circ\mathsf{M}) \leq \lambda_{\max}(\mathsf{E}) \max_{i=1,\dots,N} \mathsf{M}_{ii}.$$

This result allows us to infer about the spectrum of the kernel matrix (see [12]) and to show that with the Gaussian VSK we gain both in terms of stability and expressiveness of the kernel.

Spectral ratio for the Gaussian VSK

We now give upper and lower bounds for the Frobenius norm $\|\cdot\|_{\mathsf{F}}$ of the kernel matrix K in terms of its variably scaled setting. This turns out to be helpful when comparing the spectral ratio of the two matrices (K and K^{Ψ}).

Theorem 4.3 *Let* $\Xi = \{\boldsymbol{x}_i,\ i = 1,\dots,N\} \subseteq \Omega$ *be a set of distinct data. Let* $\psi : \Omega \longrightarrow \Lambda$ *be the scaling function for the VSK setting. Let* $\kappa : \Omega\times\Omega \longrightarrow \mathbb{R}$ *be the Gaussian kernel. Given the VSK matrix* $\mathsf{K}^{\Psi} = \mathsf{K}\circ\mathsf{K}^{\psi}$ *constructed on* Ξ *via* $\kappa^{\Psi} : \Omega\times\Omega \longrightarrow \mathbb{R}$*, we have that*

$$\|\mathsf{K}^{\Psi}\|_{\mathsf{F}} \leq \|\mathsf{K}\|_{\mathsf{F}} \leq \|\mathsf{K}\|_{\mathsf{F}}\|\mathsf{K}^{\psi}\|_{\mathsf{F}}.$$

Proof: Being the RBF $\varphi : \Omega \longrightarrow \mathbb{R}$ associated to the Gaussian kernel κ non-increasing, for $\boldsymbol{x}, \boldsymbol{y} \in \Omega$, we obtain

$$\varphi\left(\|\boldsymbol{x}-\boldsymbol{y}\|_2^2\right) \geq \varphi\left(\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \|\psi(\boldsymbol{x})-\psi(\boldsymbol{y})\|_2^2\right),$$

which in particular implies that

$$\mathsf{K}_{ij} \geq \mathsf{K}^{\Psi}_{ij} \geq 0, \quad i,j = 1,\dots,N.$$

Thus, we get

$$\|\mathsf{K}\|_{\mathsf{F}} \geq \|\mathsf{K}^{\Psi}\|_{\mathsf{F}}.$$

Moreover, since $\varphi(0) = 1$, i.e. $\mathsf{K}^{\psi}_{ii} = 1$, $i = 1,\dots,N$, we obtain

$$\|\mathsf{K}^{\psi}\|_{\mathsf{F}} \geq \sqrt{\sum_{i=1}^{N} \left(\mathsf{K}^{\psi}_{ii}\right)^2} = \sqrt{N(\varphi(0))^2} \geq 1,$$

and therefore

$$\|\mathsf{K}^{\Psi}\|_{\mathsf{F}} \leq \|\mathsf{K}\|_{\mathsf{F}} \leq \|\mathsf{K}\|_{\mathsf{F}}\|\mathsf{K}^{\psi}\|_{\mathsf{F}}.$$

□

From this theorem, we can easily infer on the spectral ratio in the VSK setting.

Corollary 4.1 *Let* $\Xi = \{\boldsymbol{x}_i,\ i = 1,\dots,N\} \subseteq \Omega$ *be a set of distinct data. Let* $\psi : \Omega \longrightarrow \Lambda$ *be the scaling function for the VSK setting. Let* $\kappa : \Omega\times\Omega \longrightarrow \mathbb{R}$ *be the Gaussian kernel, then the VSK kernel* $\kappa^{\Psi} : \Omega\times\Omega \longrightarrow \mathbb{R}$ *is more expressive than* κ*.*

Proof: Let $\mathsf{K}^{\Psi} = \mathsf{K} \circ \mathsf{K}^{\psi}$ be the VSK matrix constructed on Ξ via $\kappa^{\Psi} : \Omega \times \Omega \longrightarrow \mathbb{R}$, we have that

$$\mathrm{tr}(\mathsf{K}^{\Psi}) = \mathrm{tr}(\mathsf{K}) = N\varphi(0) = N,$$

where $\varphi : \Omega \longrightarrow \mathbb{R}$ is the RBF associated to the Gaussian kernel $\kappa : \Omega \times \Omega \longrightarrow \mathbb{R}$. Taking into account Theorem 4.3, we obtain

$$S(\mathsf{K}) = \frac{N}{\|\mathsf{K}\|_{\mathsf{F}}} \leq \frac{N}{\|\mathsf{K}^{\Psi}\|_{\mathsf{F}}} = S(\mathsf{K}^{\Psi}).$$

□

On one side, the fact that the Gaussian VSK is more expressive than the standard one tells us that the VSK-based learning might be able to deal with more complex tasks. In the next subsection, we focus on the stability of the kernel matrix.

Spectrum of the Gaussian VSK

The smallest eigenvalue of a positive definite kernel matrix is of course linked to the ill-conditioning. Moreover, given $\Xi = \{\boldsymbol{x}_i,\ i = 1, \ldots, N\} \subseteq \Omega$, the stability is also related to the separation distance

$$q_{\Xi} \coloneqq \frac{1}{2} \min_{i \neq j} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2,$$

which only depends on the data. As shown in e.g. [6], we have that

$$q_{\Xi} \leq q_{\Xi}^{\Psi},$$

where

$$q_{\Xi}^{\Psi} \coloneqq \frac{1}{2} \min_{i \neq j} \|\Psi(\boldsymbol{x}_i) - \Psi(\boldsymbol{x}_j)\|_2,$$

is the separation distance in the VSK setting. This gives the intuition of the fact that the VSKs might lead to possible improvements in terms of stability [6]. Indeed, in general, it is well-known that the smallest eigenvalue of the kernel matrix is related to the separation distance, meaning that the ill-conditioning usually grows as the separation distance decreases; refer e.g. to [27], where the authors make use of a result from [3] on the eigenvalues of distance matrices. These facts are the fruits on many studies on the so-called *trade-off* or *uncertainty principle* [36, 37], which could be summarized in a conflict between accuracy and stability.

As already mentioned, the VSKs are helpful for improving the stability, especially in view of the following property. We also refer the reader to [42, Corollary 3.1]. For a given matrix M, we focus on the 2-condition number defined as

$$\mathrm{cond}(\mathsf{M}) = ||\mathsf{M}||_2 ||\mathsf{M}^{-1}||_2.$$

Proposition 4.1 *Let $\Xi = \{\boldsymbol{x}_i,\ i = 1, \ldots, N\} \subseteq \Omega$ be a set of distinct data. Let $\psi : \Omega \longrightarrow \Lambda$ be the scaling function for the VSK setting. Let $\kappa : \Omega \times \Omega \longrightarrow \mathbb{R}$ be the Gaussian kernel. Given the VSK matrix $\mathsf{K}^{\Psi} = \mathsf{K} \circ \mathsf{K}^{\psi}$ constructed on Ξ via $\kappa^{\Psi} : \Omega \times \Omega \longrightarrow \mathbb{R}$, we have that*

$$\mathrm{cond}(\mathsf{K}^{\Psi}) \leq \mathrm{cond}(\mathsf{K}).$$

Proof: First note that, since in this case the matrix is positive definite, the condition number can be computed as

$$\mathrm{cond}(\mathsf{K}^{\Psi}) = \frac{\lambda_{\max}(\mathsf{K}^{\Psi})}{\lambda_{\min}(\mathsf{K}^{\Psi})}.$$

Moreover, from Theorem 4.2 and since the RBF $\varphi : \Omega \longrightarrow \mathbb{R}$ associated to the Gaussian kernel $\kappa : \Omega \times \Omega \longrightarrow \mathbb{R}$ is so that $\varphi(0) = 1$, i.e. $\mathsf{K}^{\psi}_{ii} = 1$, $i = 1, \ldots, N$, we obtain

$$\mathrm{cond}(\mathsf{K}^{\Psi}) = \frac{\lambda_{\max}(\mathsf{K}^{\Psi})}{\lambda_{\min}(\mathsf{K}^{\Psi})} \leq \frac{\lambda_{\max}(\mathsf{K})}{\lambda_{\min}(\mathsf{K}^{\Psi})} \leq \frac{\lambda_{\max}(\mathsf{K})}{\lambda_{\min}(\mathsf{K})} = \mathrm{cond}(\mathsf{K}).$$

□

This result turns out to be meaningful especially for the KRN-VSK approach. As a second case study, we now consider the linear kernel, which is truly popular for classification tasks.

4.2 The linear VSK

For $\boldsymbol{x}, \boldsymbol{y} \in \Omega$, the linear kernel $\kappa : \Omega \times \Omega \longrightarrow \mathbb{R}$ is given by

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}\boldsymbol{y}^{\intercal}.$$

As for the Gaussian kernel, its implementation in the variably scaled setting turns out to be trivial. We remark that the linear kernel is positive definite and thus its associated kernel matrix turns out to be positive semi-definite; see e.g. [16].

Practical implementation for the linear VSK

The linear VSK can be written as sum of matrices, indeed we have the following result.

Theorem 4.4 *Let $\Xi = \{\boldsymbol{x}_i,\ i = 1, \ldots, N\} \subseteq \Omega$ be a set of distinct data. Let $\psi : \Omega \longrightarrow \Lambda$ be the scaling function for the VSK setting. Let $\kappa : \Omega \times \Omega \longrightarrow \mathbb{R}$ be the linear kernel. Then, the VSK matrix constructed on Ξ via $\kappa^{\Psi} : \Omega \times \Omega \longrightarrow \mathbb{R}$ is given by*

$$\mathsf{K}^{\Psi} = \mathsf{K} + \mathsf{K}^{\psi},$$

where $\mathsf{K}^{\psi}_{ij} = \psi(\boldsymbol{x}_i)\psi(\boldsymbol{x}_j)^{\intercal}$, $i, j = 1, \ldots, N$.

Proof: For $\boldsymbol{x}, \boldsymbol{y} \in \Omega$ we have that:

$$\kappa^{\Psi}(\boldsymbol{x}, \boldsymbol{y}) = \begin{pmatrix} \boldsymbol{x}, & \psi(\boldsymbol{x}) \end{pmatrix} \begin{pmatrix} \boldsymbol{y}^{\intercal} \\ \psi(\boldsymbol{y})^{\intercal} \end{pmatrix} = \boldsymbol{x}\boldsymbol{y}^{\intercal} + \psi(\boldsymbol{x})\psi(\boldsymbol{y})^{\intercal},$$

and thus the kernel matrix is given by

$$\mathsf{K}^{\Psi} = \mathsf{K} + \mathsf{K}^{\psi}.$$

□

We now drive our attention towards the expressiveness of the linear VSK.

Spectral ratio for the linear VSK

Depending on the function ψ, we might have that the linear VSK is less expressive than the standard linear kernel, indeed we have the following proposition.

Proposition 4.2 *Let $\Xi = \{\boldsymbol{x}_i,\ i = 1, \ldots, N\} \subseteq \Omega$ be a set of distinct data. Let $\psi : \Omega \longrightarrow \Lambda$ be the scaling function for the VSK setting. Let $\kappa : \Omega \times \Omega \longrightarrow \mathbb{R}$ be the linear kernel. Let us suppose that the associated kernel matrix K is non-negative, i.e. so that all the entries of K are non-negative. Given the VSK matrix $\mathsf{K}^{\Psi} = \mathsf{K} + \mathsf{K}^{\psi}$ constructed on Ξ via $\kappa^{\Psi} : \Omega \times \Omega \longrightarrow \mathbb{R}$, if ψ is so that K^{ψ} is non-negative, then:*

$$\frac{\mathrm{tr}(\mathsf{K})}{\mathrm{tr}(\mathsf{K}^{\Psi})} \leq \frac{S(\mathsf{K})}{S(\mathsf{K}^{\Psi})} \leq \frac{||\mathsf{K}^{\Psi}||_{\mathsf{F}}}{||\mathsf{K}||_{\mathsf{F}}}.$$

Proof: Under our assumptions, if $\psi : \Omega \longrightarrow \Lambda$ is so that K^{ψ} is non-negative, we have that

$$\frac{\mathrm{tr}(\mathsf{K})}{\mathrm{tr}(\mathsf{K}^{\Psi})} \leq 1.$$

Moreover, since we suppose K non-negative, we get

$$\frac{||\mathsf{K}^{\Psi}||_{\mathsf{F}}}{||\mathsf{K}||_{\mathsf{F}}} \geq 1.$$

Finally, taking into account the definition of the spectral ratio, the statement follows. □

Note that the requirements of Proposition 4.2 are satisfied e.g. if $\Omega \subseteq \mathbb{R}^n_+$ and $\Lambda \subseteq \mathbb{R}^m_+$.

Spectrum of the linear VSK

Being Gramiam matrices, K^{Ψ} and K^{ψ} are positive semi-definite. Concerning the spectrum of the VSK matrix K^{Ψ}, by virtue of Weyl's inequality (see e.g. [5, Section III.2, p. 62]), we obtain that:

$$\lambda_{\min}(\mathsf{K}) \leq \lambda_{\min}(\mathsf{K} + \mathsf{K}^{\psi}) = \lambda_{\min}(\mathsf{K}^{\Psi}).$$

As for the Gaussian kernel, one can make many different choices for the function ψ. Some of them are discussed in the next section.

5 Choices for the scaling function

In the framework of approximation theory, as well as for KRNs, the choice of the scaling function can be guided by some characteristics concerning the data distribution or the underlying function that needs to be reconstructed (see e.g. [34, 35]). In the classification setting the VSKs can be seen as feature augmentation methods. More precisely, our aim is to adopt this strategy to encode possible a priori information in the kernel. Let us take N distinct data $\Xi = \{\boldsymbol{x}_i,\ i = 1, \dots, N\} \subseteq \Omega$, where $\Omega \subseteq \mathbb{R}^n$ is a bounded set. Moreover, let $\Lambda \subseteq \mathbb{R}^m$ be a bounded set, we now propose some techniques to define the scaling function of the VSK framework.

5.1 Scaling function for SVM-VSK

Depending on the task and on the available knowledge, different choices for the scaling function could be taken into account. Here, we construct the scaling function $\psi : \Omega \longrightarrow \Lambda$ as follows. Given the dataset

$$\Sigma = \{(\boldsymbol{x}_i, y_i),\ i = 0, \dots, N,\ \boldsymbol{x}_i \in \Omega,\ y_i \in \{-1, +1\}\},$$

we introduce the classes C_1 and C_2, associated to the labels $y = -1$ and $y = +1$, respectively. Let $\tilde{\boldsymbol{x}} = (\tilde{x}_1, \dots, \tilde{x}_n)$ be a new example that we need to classify. Treating the features as mutually independent, the NB classifier (see e.g. [1, 24]) computes

$$P_j(\tilde{\boldsymbol{x}}) \coloneqq P(\tilde{\boldsymbol{x}} \in C_j | \tilde{\boldsymbol{x}}) = \frac{P(C_j) \prod_{i=1}^{n} P(\tilde{x}_i | C_j)}{P(\tilde{\boldsymbol{x}})},$$

classifying

$$C(\tilde{\boldsymbol{x}}) = \operatorname{argmax}_{j=1,2} P_j(\tilde{\boldsymbol{x}}).$$

The *likelihood* $\prod_{i=1}^{n} P(\tilde{x}_i | C_j)$ and the *prior* $P(C_j)$ are typically estimated from the dataset Σ. In other cases, especially when the dataset is not too large, they could be obtained as a priori knowledge, for example by consulting the literature.

In this view, for the SVM-VSK we propose the scaling map $\Psi : \Omega \longrightarrow \tilde{\Omega}$ defined by

$$\Psi(\boldsymbol{x}) \coloneqq (\boldsymbol{x}, P_1(\boldsymbol{x})),$$

and the kernel $\kappa^{\Psi} : \Omega \times \Omega \longrightarrow \mathbb{R}$

$$\kappa^{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \coloneqq \kappa\left(\Psi(\boldsymbol{x}), \Psi(\boldsymbol{y})\right).$$

For $\boldsymbol{x} \in \Omega$, since $P_2(\boldsymbol{x}) = 1 - P_1(\boldsymbol{x})$ is correlated to $P_1(\boldsymbol{x})$, we observe that it is sufficient to consider one of the two probabilities.

Concerning the effectiveness of this scaling function $\Psi : \Omega \longrightarrow \tilde{\Omega}$ for the Gaussian VSK, we refer to the notation introduced in Theorem 4.1 and we point out that, for $\boldsymbol{x}_i, \boldsymbol{x}_j \in \Xi$,

$$\mathsf{K}^{\psi}_{ij} = \mathrm{e}^{-(P_1(\boldsymbol{x}_i) - P_1(\boldsymbol{x}_j))^2}, \quad i, j = 1, \dots, N.$$

We observe that if $P_1(\boldsymbol{x}_i) \approx P_1(\boldsymbol{x}_j)$, then $\mathsf{K}^{\psi}_{ij} \approx 1$ and so $\mathsf{K}^{\Psi}_{ij} \approx \mathsf{K}_{ij}$. Considering instead the linear VSK $\kappa^{\Psi} : \Omega \times \Omega \longrightarrow \mathbb{R}$ described in Section 4.2, we get

$$\mathsf{K}^{\psi}_{ij} = P_1(\boldsymbol{x}_i)P_1(\boldsymbol{x}_j), \quad i, j = 1, \ldots, N.$$

We remark that, according to to Proposition 4.2, with the linear VSK we construct kernels that might be less expressive than the standard ones.

For both kernels, this means that the matrices *change* according to our a priori knowledge on the dataset, leading to a different, possibly easier, learning task for SVM.

5.2 Scaling function for KRN-VSK

Here we take again N distinct data $\Xi = \{\boldsymbol{x}_i,\ i = 1, \ldots, N\} \subseteq \Omega$, where $\Omega \subseteq \mathbb{R}^n$ is a bounded set, and the associated measurements $y_i \in \mathbb{R}$, $i = 1, \ldots, N$. Moreover, let $\Lambda \subseteq \mathbb{R}^m$ be a bounded set, we now propose some ideas to define the scaling function for KRNs.

Regression networks are also used to learn longitudinal data, i.e. time series, see e.g. [8]. This allows to extrapolate the evolution of samples and consequently to give short time predictions on the dynamics of the considered process. Therefore, concerning the choice of the scaling function $\psi : \Omega \longrightarrow \Lambda$, we suppose to know the trend of data, which can be modelled via a specific class of functions, i.e. a model $\mathcal{M} : \Omega \times \mathbb{R}^l \longrightarrow \mathbb{R}$ depending on $\boldsymbol{x} \in \Omega$, and on l parameters $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_l)$. To determine $\boldsymbol{\beta}$, we compute:

$$\boldsymbol{\beta}^* = \min_{\boldsymbol{\beta} \in \mathbb{R}^l} \sum_{i=1}^{N} (y_i - \mathcal{M}(\boldsymbol{x}_i, \boldsymbol{\beta}))^2.$$

Then, one possible solution to define the function $\psi : \Omega \longrightarrow \Lambda$ is

$$\psi(\boldsymbol{x}) = \mathcal{M}(\boldsymbol{x}, \boldsymbol{\beta}^*). \tag{5.1}$$

Of course, this gives a recipe for the selection of the scaling function which is not unique. Moreover, thinking of time series, one usually disposes of other available and correlated data sampled at the same locations which can be used as additional features. For instance, when studying the evolution of a tumor mass in time, usually also other bio-markers are available and can be taken into account; see e.g. [40].

6 Numerical tests

Experiments have been carried out in PYTHON using also the scientific module scikit-learn [31] on a Intel(R) Core(TM) i7 CPU 4712MQ 2.13 GHz processor. For the classification setting, we provide a freely available software that can be downloaded at `https://github.com/emmaA89/SVM-VSK`. For KRN-VSK, we refer the reader to [16, Program 18.1, p. 340].

6.1 Tests for SVM-VSK

We present two examples of binary classification. In the first one, we consider different toy datasets of various sizes, with precise probability information concerning the features distributions, and we compare our SVM-VSK approach with standard SVM and NB classifiers. For the second experiment, we take a real dataset and we show a feature extraction strategy inspired by our framework.

In both the examples, the hyperparameters are validated by taking

$$\zeta \in \{2^{-6}, 2^{-5}, \dots, 2^{6}\},$$

$$\gamma \in \{10^{-6}, 10^{-5}, \dots, 10^{2}\}.$$

Moreover, in the validation and in the test steps, we evaluate the performance of the considered methods by means of the f_1-score, weighted with respect to the classes. We remind that the f_1-score is defined as the harmonic mean between precision and recall. More precisely, given the number of True Positive (TP), False Positive (FP) and False Negative (FN) cases,

$$\mathrm{f}_1\text{-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}},$$

where $\text{precision} = \dfrac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$ and $\text{recall} = \dfrac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$.

Test 1: Simulated datasets and a priori *knowledge*

Here we construct 12 toy datasets that differ in terms of number of features and examples. We now fix $n = 64$. Letting $\Omega \subseteq \mathbb{R}^n$, where Ω is a bounded set, they are extracted from the dataset

$$\Gamma = \{(\boldsymbol{x}_i, y_i),\ i = 1, \dots, 5000,\ \boldsymbol{x}_i \in \Omega,\ y_i \in \{-1, +1\}\},$$

where the two classes C_1 and C_2 are exactly balanced. The construction of such dataset is explained in the following steps.

1. Each class $C_j,\ j = 1, 2$, is characterized by two vectors

   $$\boldsymbol{\mu}_j = (\mu_j^1, \dots, \mu_j^n), \quad \boldsymbol{\sigma}_j = (\sigma_j^1, \dots, \sigma_j^n).$$

   More precisely, let us denote by $\mathcal{U}(a, b)$ a univariate uniform random distribution on the interval $(a, b) \subseteq \mathbb{R}$ and by $p \sim \mathcal{U}(a, b)$ a real measurement sampled from such distribution. Then, μ_j^i and σ_j^i, $i = 1, \dots, n$, $j = 1, 2$ are determined as follows:

   $$\begin{aligned}
   &\mu_1^i \sim \mathcal{U}(0, 20),\\
   &\sigma_1^i \sim \mathcal{U}(0, 2),\\
   &\mu_2^i = \mu_1^i + u^i, \text{ with } u^i \sim \mathcal{U}(0, 2),\\
   &\sigma_2^i \sim \mathcal{U}(0, 4.5).
   \end{aligned}$$

2. We denote by $\mathcal{N}(\mu, \sigma)$ the univariate normal distribution with mean μ and standard deviation σ. Let $\boldsymbol{x}_k = (x_k^1, \ldots, x_k^n)$ be an example in Ω belonging to a class C_j, $j = 1, 2$. The elements x_k^i of $\boldsymbol{x}_k \in \Omega$, $i = 1, \ldots, n$, are then randomly generated as samples of $\mathcal{N}(\mu_j^i, \sigma_j^i)$.

3. Finally, Gaussian white noise, distributed according to $\mathcal{N}(0, 1)$, is added to each feature and example.

From the so-constructed Γ, letting Ω_k be a bounded set of $\mathbb{R}^{n_k}$, $n_k \in \{2, 4, 16, 64\}$, we extract the datasets

$$\Gamma_{\boldsymbol{\xi}} = \{(\boldsymbol{x}_i, y_i),\ i = 1, \ldots, N_q,\ \boldsymbol{x}_i \in \Omega_k,\ y_i \in \{-1, +1\}\},$$

with $N_q \in \{100, 500, 2500\}$, $\boldsymbol{\xi} = (N_q, n_k)$, and preserving the balance among the two classes. Fixed N_q and n_k, we point out that the examples and the features are randomly selected. Moreover, since all combinations examples-features are taken into account, we obtain 12 different datasets.

In the following description, we fix one of the extracted datasets $\Gamma_{\boldsymbol{\xi}}$ for some value of N_q and n_k. We divide such a dataset in a training set $\Sigma_{\boldsymbol{\xi}}$ and a test set $T_{\boldsymbol{\xi}}$. These sets are so that $\text{card}(\Sigma_{\boldsymbol{\xi}}) \approx 2\text{card}(T_{\boldsymbol{\xi}})$.

In this experiment, we suppose to have a priori information and to encode it in the SVM-VSK method by means of the NB algorithm. More precisely, the NB classifier is trained considering both $\Sigma_{\boldsymbol{\xi}}$ and $\bar{\Gamma}_{\boldsymbol{\xi}}$, which is defined as the dataset containing the examples of Γ that are not in $\Gamma_{\boldsymbol{\xi}}$, i.e.

$$\bar{\Gamma}_{\boldsymbol{\xi}} \coloneqq \Gamma \setminus \Gamma_{\boldsymbol{\xi}}.$$

Therefore, in this test we compare the performances on $T_{\boldsymbol{\xi}}$ of the three methods constructed as follows.

1. The NB classifier, which is trained on $\bar{\Gamma}_{\boldsymbol{\xi}} \cup \Sigma_{\boldsymbol{\xi}}$. Given $\boldsymbol{x} = (x_1, \ldots, x_{n_k})$, we adopt the Gaussian likelihood [32]

$$P(x_i|C_j) = \frac{1}{\sqrt{2\pi\left(\sigma_j^i\right)^2}} \mathrm{e}^{-\left(\frac{x_i - \mu_j^i}{\sqrt{2}\sigma_j^i}\right)^2}.$$

   for $i = 1, \ldots, n_k$, $j = 1, 2$.

2. The standard SVM method, which is trained on $\Sigma_{\boldsymbol{\xi}}$.

3. The SVM-VSK classifier, which is trained on $\Sigma_{\boldsymbol{\xi}}$ and whose scaling map $\psi : \Omega \longrightarrow \Lambda$, constructed as explained in Section 5, considers the probabilistic outcomes of the NB classifier.

In order to tune the SVM hyperparameters ζ and γ, the latter in case of RBF kernel, we consider a 5-fold cross validation on $\Sigma_{\boldsymbol{\xi}}$.

We carry out the test for each dataset $\Gamma_{\boldsymbol{\xi}}$ and we show the obtained results in Figure 1. The proposed SVM-VSK algorithm is competitive with the best among SVM and NB methods, slightly outperforming both in some cases.

Figure 1: The f_1-score of the experiments performed on various datasets using the linear (lin.) and Gaussian kernel (RBF). The considered number of examples and features are displayed on the top.

For the Gaussian kernel, we numerically verify Corollary 4.1 by reporting in Table 1 the spectral ratios related to the matrices K and K^Ψ, obtained from the training sets $\Sigma_{\boldsymbol{\xi}}$ with $N_q = 100,\ 500,\ 2500$ and $n_k = 2$. The results numerically confirm what theoretically observed, i.e. the Gaussian VSK is more expressive than the standard one.

Moreover, for the linear kernel, we are in the hypothesis of Proposition 4.2 since in our experiments the data are normalized in $[0, 1]$. In order to numerically verify its consistency, we also evaluate the quantities involved in such proposition. They are reported in Table 2. The results show once more what theoretically observed.

N	$S(\mathsf{K})$	$S(\mathsf{K}^\Psi)$
50	1.3782 E+00	1.5756E+00
500	1.3222 E+00	1.5197E+00
2500	1.2770 E+00	1.4954E+00

Table 1: The spectral ratios of the matrices K and K^Ψ related to the normalized training sets $\Sigma_{\boldsymbol{\xi}}$, varying $N_q = 100,\ 500,\ 2500$. We set $n_k = 2$ and we considered a Gaussian kernel with $\gamma = 1$.

N	$\mathrm{tr}(\mathsf{K})/\mathrm{tr}(\mathsf{K}^\Psi)$	$S(\mathsf{K})/S(\mathsf{K}^\Psi)$	$\|\|\mathsf{K}^\Psi\|\|_\mathsf{F}/\|\|\mathsf{K}\|\|_\mathsf{F}$
50	6.1373 E-01	8.8640E-01	1.4443E+00
500	5.2890 E-01	8.8021E-01	1.6642E+00
2500	5.5705 E-01	8.8461E-01	1.5880E+00

Table 2: The ratios of the norms involved in Proposition 4.2 obtained via the linear kernel. The matrices K and K^Ψ are related to the normalized training sets $\Sigma_{\boldsymbol{\xi}}$, varying $N_q = 100,\ 500,\ 2500$ and with $n_k = 2$.

Test 2: Feature selection with a real dataset

The Wisconsin Breast Cancer Database [22, 23] consists of 699 instances described by 9 features, extracted from a digitized image of a fine needle aspirate of a breast mass. The task consists in predicting if the mass is benign or malignant. From the original dataset, we exclude 16 instances that present missing values. The two classes are not equally distributed, presenting 444 benign instances and 239 malignant instances.

In this section, we propose a feature extraction method directly inspired by the presented variably scaled setting, which can be used as an alternative to other possible expensive feature extraction algorithms. To this aim, at first, we divide the dataset into a training set, consisting of 226 benign and 116 malignant cases, and a test set, which is composed by 218 benign and 123 malignant cases.

Then, we compare the performances on the test set of the following four methods.

1. A NB classifier with Gaussian likelihood.

2. A standard SVM classifier, whose hyperparameters ζ and γ (in the Gaussian case) are validated by means of 5-fold cross validation on the training set.

3. A SVM classifier constructed after a feature selection process, as explained in what follows.
   Analyzing the resulting weights of the SVM classifier (in the linear case), we can rank the features by their influence in the classification; see e.g. [17]. Then, we choose the $\bar{n}$ more relevant features, here we fix $\bar{n} = 2$, and we consequently reduce our training and test sets by restricting to the two most relevant features. Finally, we take both linear and Gaussian kernels, we train a SVM classifier via 5-fold cross validation on the reduced training set and we evaluate the results on the reduced test set.
   We denote this method with SVM-Selection (SVM-S).

4. A SVM classifier constructed after a VSK-like feature extraction process, as described in the following lines.
   We randomly select $\bar{n}-1$ features (here $\bar{n} = 2$). The training set restricted to the remaining 8 features is used to train a Gaussian NB classifier. Reduced training and test sets are obtained by juxtaposing the previously

selected $\bar{n} - 1$ features to the probabilistic output of the NB classifier. Then, we take both linear and Gaussian kernels, we train a SVM classifier via 5-fold cross validation on the reduced training set and we evaluate the results on the reduced test set.
We denote this method with SVM-Extraction (SVM-E).

We point out that both SVM-S and SVM-E consider reduced training and test sets that are characterized by the same number of features $\bar{n}$. Moreover, the SVM-E presents some advantages in terms of computational complexity with respect to SVM-S, since training an auxiliary NB classifier to perform feature extraction is cheaper than training a SVM classifier to carry out the feature selection.

In Table 3, we present the results obtained considering the SVM, NB and SVM-S methods. In Table 4, we report the results concerning the SVM-E algorithm. For completeness, we vary the randomly selected feature, taking into account all the possibilities.

	Linear		Gaussian	
NB	SVM	SVM-S	SVM	SVM-S
0.965	0.968	0.959	0.965	0.953

Table 3: The f_1-score for the Wisconsin Breast Cancer Database via the SVM, NB and SVM-S methods.

Random Feature	Linear	Gaussian
1	0.965	0.965
2	0.962	0.968
3	0.959	0.977
4	0.962	0.965
5	0.965	0.965
6	0.965	0.962
7	0.965	0.962
8	0.959	0.956
9	0.968	0.965

Table 4: The f_1-score for the Wisconsin Breast Cancer Database via the SVM-E method.

We observe that the best score is achieved by the SVM-E algorithm. Moreover for this dataset, we point out that such a method prefers the Gaussian kernel with respect to the linear one, while the standard SVM and SVM-S obtain better classification scores when the linear kernel is considered.

6.2 Tests for KRN-VSK

As an example for KRNs, we focus on the Italian data of the 2020 COVID19 pandemic. The task we consider consists in learning the longitudinal data, i.e. $\Omega \subseteq \mathbb{R}$, of people that in Italy were hospitalized as Intensive Care Unit (ICU) patients from 24/02/2020 to 26/04/2020. The dataset, provided by the "Dipartimento della Protezione Civile", is available at
`https://github.com/pcm-dpc/COVID-19/tree/master/dati-andamento-nazionale`.

The dataset Γ consists of 63 samples and it is divided as follows. The first 58 days are used as training set Σ and we test the model on the last $t = 5$ days, $\tilde{x}_i,\ i = 1, \dots, t$. Referring to Subsection 3.2 we take the set of kernel centres Z as the set of available data in Ξ and we focus on the Gaussian kernel. The feature augmentation strategy outlined in (5.1) is carried out considering $\mathcal{M} : \Omega \times \mathbb{R} \longrightarrow \mathbb{R}$ defined as

$$\mathcal{M}(x, \beta) = \mathrm{e}^{-\beta|x-\bar{p}|},$$

where $\bar{p} = 42$ is the peak of the considered time series. The model $\mathcal{M}$ is constructed on Σ.

Furthermore, we encode into the kernel also other available data. Specifically, we take the total number of COVID19 infected (included death and recovered people), the daily number of new infected and the total number of infected (excluded death and recovered people). Of course, this selection of the scaling function means that we are adding a priori knowledge to the selected time series. Therefore, the scaling function ψ is so that $\psi : \Omega \longrightarrow \Lambda$, where $\Lambda \subseteq \mathbb{R}^4$.

To analyze the performances of the variably scaled setting, we take the Gaussian kernel and we compute the condition number of the kernel matrix and the Rounded Mean Error (RME). Precisely, since hospitalized patients are involved in the dynamics, letting the Mean Error

$$\mathrm{ME} = \frac{1}{t} \sum_{i=1}^{t} |y_i - A(\tilde{x}_i)|,$$

where A is a decision function as defined in Subsection 3.2 obtained via classical of variably scaled kernels, the RME is defined as

$$\mathrm{RME} = \begin{cases} \lfloor \mathrm{ME} \rfloor, & \text{if} \quad \mathrm{ME} - \lfloor \mathrm{ME} \rfloor \leq 0.5, \\ \lceil \mathrm{ME} \rceil, & \text{if} \quad \mathrm{ME} - \lfloor \mathrm{ME} \rfloor > 0.5. \end{cases}$$

In the first experiments, we set the parameter $\nu = 0$. We remark that for regression networks the selection of the shape parameter pays a crucial role. Therefore, to make a fair comparison between classical and VSK regression networks, we report the condition numbers and the RME for 200 values of the shape parameter γ in the interval $[0.5, 20]$. The results are reported in Figure 2. We observe that the computation carried out via VSKs is characterized by a lower condition number of the kernel matrix, as theoretically observed in Proposition 4.1. For such experiment, this directly reflects on the accuracy of

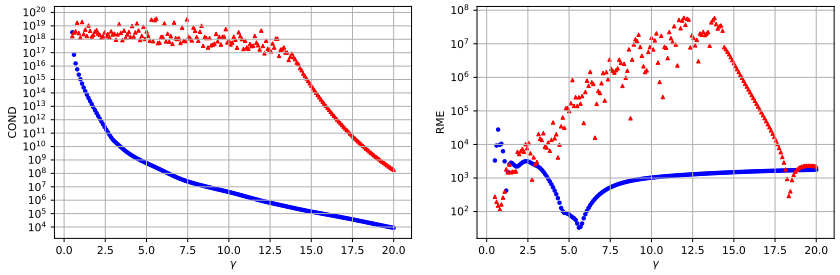

Figure 2: Left: the condition numbers for different values of the shape parameter of the classical kernel and VSK matrix denoted by red triangles and blue dots, respectively. Right: the RME for different values of the shape parameter of the classical KRN and KRN-VSK methods denoted by red triangles and blue dots, respectively. Both plots are in semi-logarithmic scale and obtained by considering the normalized dataset.

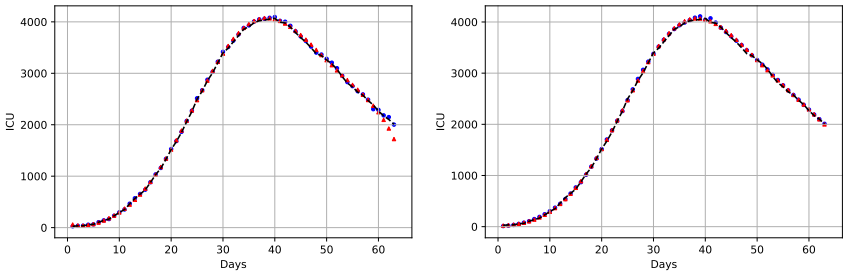

Figure 3: The ICU patients curves reconstructed via KRN and KRN-VSK denoted by red triangles and blue dots, respectively. We fix $\nu = 0$ and $\nu = 1\text{e}-04$, left and right respectively. The true solution is plotted with black solid line.

the computation, meaning that the *safe* interval for the shape parameter γ is larger than for the classical method (see Figure 2, right).

In Figure 3, we report two graphical results corresponding to $\nu = 0$ and $\nu = 1\text{e} - 04$, left and right respectively. In both cases we take the *optimal* shape parameter γ^*, meaning that it leads to the smallest RME, in the same framework of Figure 2 (right). The associated RME is shown in Table 5. We note that, the VSK setting outperforms the classical method for $\nu = 0$, while for $\nu = 1\text{e} - 04$ the two approximations are comparable.

Method	ν	
	$\nu = 0$	$\nu = 1\text{e}-04$
KRN	116	13
KRN-VSK	33	9

Table 5: The RME for the optimal shape parameter by using KRN and KRN-VSK in reconstructing the ICU curves.

7 Conclusions and future work

We presented an original approach for learning issues via VSKs. The proposed methods turn out to be flexible and easy to implement. For KRNs, the use of VSKs takes advantage of being stable and for classification of merging the probabilistic features of NB and the geometric ones of SVM. This results in effective algorithms that can be used for many tasks. Applications to real datasets show the effectiveness of our approach.

Work in progress consists in extending this concept for support vector regression and as well as for greedy methods [2, 45].

Acknowledgements

We sincerely thank the reviewers for helping us to significantly improve the manuscript. This research has been accomplished within Rete ITaliana di Approssimazione (RITA) and partially funded by GNCS-INδAM, by the European Union's Horizon 2020 research and innovation programme ERA-PLANET, grant agreement no. 689443, via the GEOEssential project and by the ASI - INAF grant "Artificial Intelligence for the analysis of solar FLARES data (AI-FLARES)".

References

[1] C.C. Aggarwal, *Data Classification: Algorithms and Applications*, Boca Raton, FL, USA, CRC Press, 2014.

[2] M. Aminian Shahrokhabadi, A. Neisy, E. Perracchione, M. Polato, *Learning with subsampled kernel-based methods: Environmental and financial applications*, Dolomites Res. Notes Approx. **12** (2019), 17–27.

[3] K. Ball, *Eigenvalues of Euclidean distance matrices*, J. Approx. Theory **68** (1992), 74–82.

[4] P.L. Bartlett, S. Mendelson *Rademacher and Gaussian complexities: risk bounds and structural results*, Journal of Machine Learning Research **3** (2002), 463–482.

[5] R. Bhatia, *Matrix Analysis*, Springer-Verlag, New York, 1997.

[6] M. Bozzini, L. Lenarduzzi, M. Rossini, R. Schaback, *Interpolation with variably scaled kernels*, IMA J. Numer. Anal. **35** (2015), 199–219.

[7] D.T. Bui, B. Pradhan, O. Lofman, I. Revhaug, *Landslide susceptibility assessment in Vietnam using support vector machines, decision tree and Naïve Bayes models*, Math. Probl. Eng. 1–26.

[8] R. Campagna, C. Conti, S. Cuomo, *Smoothing exponential-polynomial splines for multiexponential decay data*, Dolomites Res. Notes Approx. **12** (2019), 86–100.

[9] H. Daumé, *Frustratingly easy domain adaptation*. In: Association for computational linguistics (ACL), 2007.

[10] S. De Marchi, W. Erb, F. Marchetti, E. Perracchione, M. Rossini, *Shape-Driven Interpolation with Discontinuous Kernels: Error Analysis, Edge Extraction and Applications in MPI*, SIAM J. Sci. Comput. **42** (2020), B472–B491.

[11] S. De Marchi, F. Marchetti, E. Perracchione, *Jumping with Variably Scaled Discontinuous Kernels (VSDKs)*, to appear on BIT Numerical Mathematics, 2020, `https://doi.org/10.1007/s10543-020-00800-9`.

[12] B. Diederichs, A. Iske, *Improved estimates for condition numbers of radial basis function interpolation matrices*, J. Approx. Theory **238** (2019), 38–51.

[13] M. Donini, F. Aiolli, *Learning deep kernels in the space of dot product polynomials*, Machine Learning, **106** (2017), 1245–1269.

[14] N. El Karoui, *The spectrum of kernel random matrices*, Ann. Statist. **38** (2010), 1–50.

[15] G.E. Fasshauer, *Meshfree Approximations Methods with* Matlab, World Scientific, Singapore, 2007.

[16] G.E. Fasshauer, M.J. McCourt, *Kernel-based Approximation Methods Using* Matlab, World Scientific, Singapore, 2015.

[17] H. Hoffmann, *Kernel PCA for novelty detection*, Pattern Recognition, **40** (2007), 863–874.

[18] R.A. Horn, F. Zhang, *Bounds on the spectral radius of a Hadamard product of nonnegative or positive semidefinite matrices*, Electron. J. Linear Algebra **20** (2010), 90–94.

[19] K.I. Kim, K. Jung, H.J. Kim, *Face recognition using kernel principal component analysis*, IEEE Signal Processing Letters, **9** (2002), 40–42.

[20] E. Larsson, B. Fornberg, *Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions*, Comput. Math. Appl. **49** (2005), 103–130.

[21] W. Li, L. Duan, D. Xu, I.W. Tsang, *Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation*, IEEE Trans. Pattern. Anal. Mach. Intell. **36** (2014), 1134–1148.

[22] O.L. Mangasarian, W. Nick Street, W.H. Wolberg, *Wisconsin Breast Cancer Database*, UCI Machine Learning Repository, `http://archive.ics.uci.edu/ml`, University of Wisconsin, 1991.

[23] O. L. Mangasarian, W. H. Wolberg, *Cancer diagnosis via linear programming*, SIAM News, **106** (1990), 1–18.

[24] M.E. Maron, *Automatic Indexing: An Experimental Inquiry*, Journal of the ACM. **8** (1961), 404–417.

[25] J. Mercer, *Functions of positive and negative type and their connection with the theory of integral equations*, Phil. Trans. Royal Society **209** (1909), 415–446.

[26] B. Muquet, Z. Wang, G. B. Giannakis, M. De Courville, P. Duhamel, *Cyclic prefixing or zero padding for wireless multicarrier transmissions?* IEEE Transactions on Communications, vol. 50, no. 12, 2136–2148, 2002.

[27] F.J. Narcowich, J.F. Ward, *Norm estimates for the inverses of a general class of scattered-data radial-function interpolation matrices*, J . Approx. Theory **69** (1992), 84–109.

[28] J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.

[29] M.J.L. Orr, *Introduction to radial basis function networks*, Tech. rep., University of Edinburgh, Centre for Cognitive Sciences, 1996.

[30] B. Pang, B. Lee, S. Vaithyanathan, *Thumbs up? Sentiment Classification Using Machine Learning Techniques*, In Proc. of EMNLP 2002, 79–86.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, *Scikit-learn: Machine Learning in Python*, Journal of Machine Learning Research, **12** (2011) 2825–2830.

[32] J. Ren, S. D. Lee, X. Chen, B. Kao, R. Cheng, D. Cheung, *Naive Bayes classification of uncertain data*, in Proc. 9th IEEE Int. Conf. Data Mining (ICDM), 2009, 944–949.