Technical report : SVM in Krein spaces

Gaëlle Loosli, Cheng Soon Ong, Stephane Canu

SVM in Kreĭn spaces

Gaëlle Loosli , gaelle@loosli.fr
Clermont Université, Université Blaise Pascal
CNRS, UMR 6158
LIMOS, Aubière, France
Cheng Soon Ong , chengsoon.ong@unimelb.edu.au
National ICT Australia, Melbourne, Australia.
Stéphane Canu , scanu@insa-rouen.fr
Laboratoire LITIS - EA 4108
INSA de Rouen
Saint-Étienne-du-Rouvray, France

Abstract

Support vector machines (SVM) and kernel methods have been highly successful in many application areas. However, the requirement that the kernel is symmetric positive semidefinite, Mercer's condition, is not always verified in practice. When it is not, the kernel is called indefinite. Various heuristics and specialized methods have been proposed to address indefinite kernels, from simple tricks such as removing negative eigenvalues, to advanced methods that de-noise the kernel by considering the negative part of the kernel as noise. Most approaches aim at correcting an indefinite kernel in order to provide a positive one.

We propose a new SVM approach that deals directly with indefinite kernels. In contrast to previous approaches, we embrace the underlying idea that the negative part of an indefinite kernel may contain valuable information. To define such a method, the SVM formulation has to be adapted to a non usual form: the stabilization. The hypothesis space, usually a Hilbert space, becomes a Kreĭn space. This work explores this new formulation, and proposes two practical algorithms (ESVM and KSVM) that outperform the approaches that modify the kernel. Moreover, the solution depends on the original kernel and thus can be used on any new point without loss of accuracy.

1 Introduction

Kernel methods are now widely used, even in industrial applications. In many applications fields, a major open question regards the design of a good kernel, able to capture precisely the proximity of the data instances. This is particularly true when data are represented by complex structures, such as graphs. While many kernels exist, significant effort has to be spent to check or prove the validity of any proposed kernel. Indeed, according to Mercer's theorem, to be valid, a kernel needs to be symmetric and positive definite. It is not always easy to obtain a true Mercer kernel, and quite often in practice, indefinite kernels are successfully used (Cortes et al., 2003; Collins and Duffy, 2001; Haasdonk and Burkhardt, 2007)). Some authors also study kernels that are positive definite with high probability (Boughorbel et al., 2004).

This raises several questions: First, even though there are obvious benefits in terms of optimization methods when kernels are positive (semi)definite, what is the learning interpretation of such a constraint? Second, admitting the assumption that an indefinite kernel is actually what is needed to learn some specific data, how can it be performed *correctly*? And finally, admitting it is possible to solve the learning problem with an indefinite kernel, without removing or distorting its negative part, is it possible to evaluate the added value of such kernels? These questions do not have straightforward answers. Many authors provide algorithms with indefinite kernels (Hsuan-Tien and Chih-Jen, 2003; Chen et al., 2009; Haasdonk and Pekalska, 2008; Pekalska and Haasdonk, 2009; Gu and Guo, 2012), emphasizing on the need for handling such kernels. Most of time, there are some limits on the *indefiniteness* of the kernel (not too many or not too large negative eigenvalues for instance).

In this paper, we focus on a Support Vector Machine (SVM) like algorithm with indefinite kernels, although several parts could be applied to different kernel methods. The starting point is the definition of the SVM when using an indefinite kernel. In this context, the set of hypotheses induced by the kernel is not a RKHS (Reproducing Kernel Hilbert Space) but a RKKS (Reproducing Kernel Kreĭn Space) (Haasdonk, 2005; Ong et al., 2004; Hassibi et al., 1999). This difference has a great impact on the definition of the system solved to obtain the SVM solution: from a standard quadratic minimization system, the SVM problem becomes a highly non standard quadratic stabilization system. This means that the objective function, usually seen as a cost, should not be minimized but stabilized. Unfortunately, optimization literature has focused on minimization or maximization, and there has been

a dearth of approaches to solve stabilization problems.

For completeness, we review the mathematics behind optimizing SVMs (Section 2). The presented work aims at understanding what this stabilization problem is and linking it to known optimization forms. We propose an SVM like stabilization problem for classification with indefinite kernels in Section 3 and review the relevant literature for Reproducing Kernel Kreĭn Space. We propose two algorithms for solving the stabilization problem: an exact solution using eigen-decomposition (Section 4) called ESVM and an approximate solution using an active set approach (Section 5) called KSVM. To be able to evaluate the proposed algorithms, we compare them to existing approaches that deal with indefinite kernels: Relevance Vector Machines (Tipping, 2001), that directly use in kernel matrix, and IndefiniteSVM (Luss and d'Aspremont, 2009; Ying et al., 2009; Chen and Ye, 2008), that considers the negative part as noise. Some well known heuristics are also taken into account, like the modification of the spectrum of the kernel matrix: the negative eigenvalues are either cut to zero or set to their absolute values. It is also possible to translate the whole spectrum until all eigenvalues are positive (Muñoz and de Diego, 2006). The experiments (Section 6) show that keeping the information contained in the negative part of the kernel can improve the learning ability of the SVM, which is in favor of the assumption that an indefinite kernel is not necessarily a noisy kernel, and it is worth exploiting it.

2 A Brief Mathematical Review of SVMs and Kernels

The mathematical properties of SVMs with kernels lie at the intersection of several useful results from various areas (Schölkopf and Smola, 2002). In this section, we briefly review these properties with the aim of precisely identifying the difficulty of generalizing SVMs to indefinite kernels.

2.1 Geometric Margin

The geometric basis of SVMs are derived by considering the distance between points to the hyperplane. The distance between the closest point to the hyperplane is called the margin, given by $\frac{1}{\|w\|}$, where w is the normal vector to the hyperplane. Based on regularization theory, margin maximization has been shown to have many learning theoretic benefits (Steinwart and Christmann, 2008).

2.2 Reproducing Kernel Hilbert Spaces

It turns out that for the purposes of learning from finite data, the inner products corresponding to general Hilbert spaces are not suitable. For example in L_2, two functions which differ on a finite set of points are indistinguishable since the finite set of points has measure zero. One more restricted space of inner products which has been successfully used in machine learning is the Reproducing Kernel Hilbert Space (RKHS). There are two equivalent definitions of RKHS, which provide different viewpoints of this highly useful inner product space (see for instance Aronszajn, 1950).

The first definition of RKHS is based on the notion of reproducing kernel. Let $\mathcal{X}$ be a nonempty index set, and $\mathcal{H}$ a Hilbert space of real valued functions defined on $\mathcal{X}$ and endowed with its inner product $\langle \cdot, \cdot \rangle$.

Definition 2.1 (Reproducing kernel) *A function* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *is a reproducing kernel of* $\mathcal{H}$ *if it verifies the following properties:*

1. $\forall x \in \mathcal{X},\ k(x, x')$ *as a function of* x' *belongs to* $\mathcal{H}$,
2. *(the reproducing property)* $\forall x \in \mathcal{X}, \forall f \in \mathcal{H}, \quad \langle f, k(x, \cdot) \rangle = f(x)$.

Reproducing kernel Hilbert spaces are then defined as Hilbert spaces which possesses a reproducing kernel. In this first definition, the notion of reproducing kernel is defined by using the inner product of a Hilbert space and thus appears to be dependent on this notion. But this is not the case. To highlight the nature of RKHS and its associated kernel, another (equivalent) definition is needed. For a given Hilbert space $\mathcal{H}$, a necessary and sufficient condition for the existence (and unicity) of an associated reproducing kernel is that for every x of the index set, $f(x)$ is a continuous functional of f running through $\mathcal{H}$ (using the Riesz representation theorem. See for instance Akhiezer and Glazman, 1993, Section 16). This leads to a second definition of a RKHS.

Definition 2.2 (Reproducing kernel Hilbert spaces) *A reproducing kernel Hilbert space is a Hilbert space* $\mathcal{H}$ *for which at each* $x \in \mathcal{X}$ *the evaluation functional* δ_x,

$$\begin{aligned} \delta_x : \quad \mathcal{H} &\longrightarrow \mathbb{R} \\ f &\longmapsto \delta_x(f) = f(x) \end{aligned}$$

is continous.

This definition is more general than the previous one in the sense that the word "Hilbert" could have been replaced by "Kreĭn". We will define the analogous concept of reproducing kernel Kreĭn spaces in Section 3.1.

2.3 Representer Theorem

When performing optimization in a RKHS in the regularized empirical risk setting, the optimal solution can be found as a linear combination of a finite number of basis functions, regardless of the dimensionality of the space $\mathcal{H}$. This result is known as the representer theorem.

Theorem 2.1 *(Theorem 4.2 in Schölkopf and Smola, 2002) Let* $\Omega : [0, \infty) \longrightarrow \mathbb{R}$ *be a strictly monotonic increasing function,* $\mathcal{X}$ *a set, and* $\ell_h : (\mathcal{X} \times \mathbb{R}^2)^\ell \longrightarrow \mathbb{R} \cup \{\infty\}$ *a loss function. Then each minimizer* $f \in \mathcal{H}$ *of the general regularized risk*

$$\ell_h((x_1, y_1, f(x_1)), \ldots, (x_\ell, y_\ell, f(x_\ell))) + \Omega(\|f\|_{\mathcal{H}})$$

admits a representation of the form

$$f(x) = \sum_{i=1}^{\ell} \alpha_i k(x_i, x),$$

where k *is the reproducing kernel of* $\mathcal{H}$*, and* $\alpha_i \in \mathbb{R}$ *for all* $i = 1, \ldots, \ell$.

2.4 SVM as a projection

The success of SVMs in numerous application areas is partly due to the availability of stable, efficient and accurate numerical algorithms. The reason for this is that the problem of margin maximization in RKHS can be cast as a quadratic program, which is a special case of convex optimization problems. More precisely, the SVM problem can be formulated as follows. Let $x_i \in \mathcal{X}^d, i \in \{1, \ldots, \ell\}$ be ℓ training points in d dimensions, along with their label $y_i \in \{-1, 1\}$ representing the class each point belongs to in a classification problem. For a given τ, SVM is the solution of the following quadratic program (QP):

$$\begin{cases} \min\limits_{f \in \mathcal{H}, b \in \mathbb{R}} & \frac{1}{2}\|f\|_{\mathcal{H}}^2 \\ \text{s.t.} & \sum_{i=1}^{\ell} \max\big(0, 1 - y_i(f(x_i) + b)\big) \leq \tau \ . \end{cases} \tag{1}$$

This QP can be seen as the problem of retrieving the orthogonal projection of the null function in $\mathcal{H}$ onto the convex feasible set $\mathcal{S} = \big\{ f \in \mathcal{H} | \sum_{i=1}^{\ell} max\big(0, 1 - y_i(f(x_i) + b)\big) \leq \tau \big\}$. Projection is a well known regularization mechanism (see for instance Hofmann et al., 2007, and related references), allowing to choose a unique and stable solution from the feasible set $\mathcal{S}$. Convex optimization problems have the desirable property that every local minimum is a

global minimum. One of the often cited drawbacks of SVMs with indefinite kernels is that is results in non-convex optimization problems (Haasdonk, 2005; Luss and d'Aspremont, 2009). However, as we will see in the following section, this projection regularization principle can be defined when dealing with undefined kernels, leading to a stationarization problem instead of minimization.

3 A stabilization quadratic program to solve SVM in Kreĭn spaces

In this section, the Reproducing Kernel Kreĭn Space (RKKS) is briefly introduced and the stabilization system to be solved to train SVM in Kreĭn space is proposed. It was shown in Ong et al. (2004) that from the function space point of view, positive semidefiniteness is *not* a requirement, and in fact the representer theorem is also valid for RKKS.

3.1 Reproducing Kernel Kreĭn Space

Kreĭn spaces are indefinite inner product spaces endowed with a Hilbertian topology. The key difference from Hilbert spaces is in the positiveness axiom no longer required for Kreĭn Space.

Definition 3.1 (Inner Product, Bognár 1974) *Let $\mathcal{K}$ be a vector space on the scalar field. An inner product $\langle .,. \rangle_{\mathcal{K}}$ on $\mathcal{K}$ is a bilinear form where for all $f, g, h \in \mathcal{K}, \alpha \in I\!R$:*

- $\langle f, g \rangle_{\mathcal{K}} = \langle g, f \rangle_{\mathcal{K}}$
- $\langle \alpha f + g, h \rangle_{\mathcal{K}} = \alpha \langle f, h \rangle_{\mathcal{K}} + \langle g, h \rangle_{\mathcal{K}}$
- $\langle f, g \rangle_{\mathcal{K}} = 0, \quad \forall g \in \mathcal{K} \implies f = 0$

Definition 3.2 (Kreĭn space, Azizov and Iokhvidov 1989) *An inner product space $(\mathcal{K}, \langle .,. \rangle_{\mathcal{K}})$ is a Kreĭn space if there exists two Hilbert spaces $\mathcal{H}_+, \mathcal{H}_-$ spanning $\mathcal{K}$ such that*

- $\forall f \in \mathcal{K}, f = f_+ + f_-$, *where* $f_+ \in \mathcal{H}_+$ *and* $f_- \in \mathcal{H}_-$
- $\forall f, g \in \mathcal{K}, \langle f, g \rangle_{\mathcal{K}} = \langle f_+, g_+ \rangle_{\mathcal{H}_+} - \langle f_-, g_- \rangle_{\mathcal{H}_-}$

If $\mathcal{H}_+$ and $\mathcal{H}_-$ are RKHS, $\mathcal{K}$ is a reproducing kernel Kreĭn spaces (RKKS). Definition 2.2 can be adapted to define RKKS (replace Hilbert by Kreĭn). In this case the uniqueness of the functional decomposition (the nature of the RKHSs $\mathcal{H}_+$ and $\mathcal{H}_-$) is not guaranteed. In (Ong et al., 2004, Proposition 6), the reproducing property is shown: in a RKKS $\mathcal{K}$, there is a unique symmetric $k(x, x')$ with $k(x, .) \in \mathcal{K}$ such that the reproduction property holds (for all $f \in \mathcal{K}, \langle f, k(x, .)\rangle_{\mathcal{K}} = f(x)$) and $k = k_+ - k_-$ where k_+ and k_- are the reproducing kernels of the RKHSs $\mathcal{H}_+$ and $\mathcal{H}_-$. Furthermore, to any symmetric nonpositive kernel k that can be decomposed as the difference of two positive kernels k_+ and k_-, it can be associated a RKKS.

3.2 SVM in RKKS

The definition of a proper SVM in RKKS requires adaptation from the classical SVM in RKHS given by (1) since the norm $\|f\|$ is not defined in Kreĭn spaces. However, as previously remarked, the minimization of a norm can be seen as a projection. This interpretation in terms of projection still hold in Kreĭn spaces and can be used as a regularization mechanism. This allows to define SVM in RKKS (as it can be in Hilbert spaces) as the orthogonal projection of the null element onto the convex feasible set $\mathcal{S} = \{f \in \mathcal{H} | \sum_{i=1}^{\ell} \max\big(0, 1 - y_i(f(x_i) + b)\big) \leq \tau\}$. As claimed in Hassibi et al. (1999, section 2.4 p 40), in Hilbert space, projections *extremize* certain quadratic forms while in Kreĭn spaces we can in general only assert that projections *stationarize* such quadratic form. In our case, this quadratic form is $\langle f, f\rangle_{\mathcal{K}}$, leading to the following formulation of indefinite SVM in RKKS

$$\left\{ \begin{array}{ll} \underset{f\in\mathcal{K}}{\text{stat}} & \frac{1}{2} \langle f, f\rangle_{\mathcal{K}} \\ \text{s.t.} & \sum_{i=1}^{\ell} \max\big(0, 1 - y_i(f(x_i) + b)\big) \leq \tau \ . \end{array} \right. \tag{2}$$

where *stat* means *stationarize*. This stationary point may be either a minimum, a maximum or a saddle point. We show in section 4.1 that it is in fact a saddle point, which will allow us a rewrite the problem conveniently. Indeed, minimization is not the wanted here since in a RKKS $\langle f, f\rangle_{\mathcal{K}} = \langle f_+, f_+\rangle_{\mathcal{H}_+} - \langle f_-, f_-\rangle_{\mathcal{H}_-}$ so that f_- can be chosen to make $\langle f, f\rangle_{\mathcal{K}}$ arbitrarily negative.

The literature on convex optimization (Boyd and Vandenberghe, 2004; Rockafellar, 1996) has focused on the solution of minimization or maximization problems. But the optimization problem required for indefinite SVMs involves a stationary point condition, which has not received much study. Interestingly, all three problems (minimization, maximization and stabilization) has the same first order conditions of optimality. To characterize the

solutions of indefinite SVM in RKKS given by (2), we find it useful to define $J(f)$ the following loss function:

$$J(f) = J_1(f) + J_2(f) \quad \text{with} \quad \begin{cases} J_1(f) = \frac{1}{2}\langle f, f\rangle_{\mathcal{K}} \\ J_2(f) = C\sum_{i=1}^{\ell} \max\big(0, 1 - y_i(f(x_i) + b)\big) \end{cases} \tag{3}$$

This function $J(f)$ is non convex and non differentiable: $J_1(f)$ is differentiable but non convex while $J_2(f)$ is convex but non differentiable. In this case, if f^* is a local stationary point (minimum, maximum or saddle point) of the cost function $J(f)$, then it verifies the inclusion

$$0 \in \partial_c J(f^*)$$

where ∂_c is the Clarke subdifferential (Clarke, 1989). The Clarke subdifferential is the generalization of the subdifferential to non convex functions. It is defined for locally Lipschitz functions h as the convex hull of some generalized gradient and more precisely

$$\partial_c h(\beta) = \big\{ g \in I\!R^p \quad | \quad g^\top d \leq D_c h(\beta, d) \quad \forall d \in I\!R^p \big\}$$

where $D_c h(\beta, d)$ denotes the Clarke directional derivative function of h at point β in direction d defined by

$$D_c h(\beta, d) = \lim_{\epsilon \to 0_+} \sup_{\delta \to \beta} \frac{h(\delta + \epsilon d) - h(\delta)}{\epsilon}$$

Note that for J it coincides with the usual directional derivative. To calculate $\partial_c J$ it is worth noticing that, for all f, function J can be split as the sum of two terms $J = J_1 + J_2$ (eq. 3), J_1 being strictly differentiable and J_2 being convex. In this case (see Clarke, 1989, proposition 2.3.3 corollary 1) the $0 \in \partial_c J(f^*)$ condition comes out as

$$f^* \text{is a local stationary point} \Rightarrow -\nabla_f J_1(f^*) \in \partial J_2(f^*) \tag{4}$$

where $\nabla_f J_1(f^*)$ is the gradient of function J_1 at point f^* and $\partial J_2(f^*)$ is the subdifferential of the convex function J_2 at point f^*.

Let apply this:

$$\begin{cases} \nabla_f J_1(f^*) = f(.) \\ \partial_f J_2(f^*) = -C\sum_{i=1}^{\ell} \beta_i y_i k(x_i, .) \quad \text{with} \quad \begin{cases} \beta_i = 0 & \text{if} \quad 1 - y_i\big(f(x_i) + b\big) < 0 \\ 0 < \beta_i < 1 & \text{if} \quad 1 - y_i\big(f(x_i) + b\big) = 0 \\ \beta_i = 1 & \text{if} \quad 1 - y_i\big(f(x_i) + b\big) > 0 \end{cases} \end{cases} \tag{5}$$

Equation 4 states that there exists a vector β^* such that $\nabla_f J_1(f^*) + \partial J_2(f^*) = 0$, hence

$$f^*(.) = C \sum_{i=1}^{\ell} \beta_i^* y_i k(x_i, .). \tag{6}$$

The solution characterized by the vector β^* may not be unique. Among those solutions, to solve the SVM problem in the Kreĭn space, we want the one that stabilizes $J_1(f)$.

This framework is the proper way to interpret and write the previously proposed system to solve SVM in a RKKS (Loosli and Canu, 2011):

$$\begin{cases} \underset{f,b,\xi}{\text{stab}} & \frac{1}{2}\langle f, f\rangle_{\mathcal{K}} + C \sum_{i=1}^{\ell} \xi_i & \\ \text{st} & y_i(f(x_i) + b) \geq 1 - \xi_i & \forall i \in [1..\ell] \\ & \xi_i \geq 0 & \forall i \in [1..\ell] \end{cases} \tag{7}$$

where *stab* means *stabilize*.

3.3 Stabilization versus minimization

The intuition behind the stabilization problem is not straight forward. It requires to think about the meaning of the negative part of the space. A very interesting viewpoint is introduced in Laub and Müller (2004), arguing that (in the context of feature discovery), *the negative eigenvalues can code for relevant structure in the data*. One of the striking examples is on the MNIST database: the projection of digits onto the first 2 positive eigendirections gathers them according to their shape (i.e. their labels), while the projection onto the 2 last negative eigendirections gathers them according to the stroke weight (which is not relevant for classification but is still relevant information). This work clearly shows the interest of keeping the negative subspace information. It also gives hints on the meaning of the stabilization: when optimizing a standard SVM, one aims at minimizing the variance (the cost function) brought by the kernel matrix (while performing correct separation). In the stabilization form of the SVM, it is the same idea, except that part of this variance comes negatively through the negative subspace of the Kreĭn space. It makes sense then to maximize this negative variance while minimizing the positive usual variance, which leads to a stabilization problem.

Figure 1 illustrates the effect of trying to minimize function $J(f)$ instead of stabilizing it when the kernel lies in a Kreĭn space. It shows on a very simple problem (2 points) which solutions are found when stabilizing and

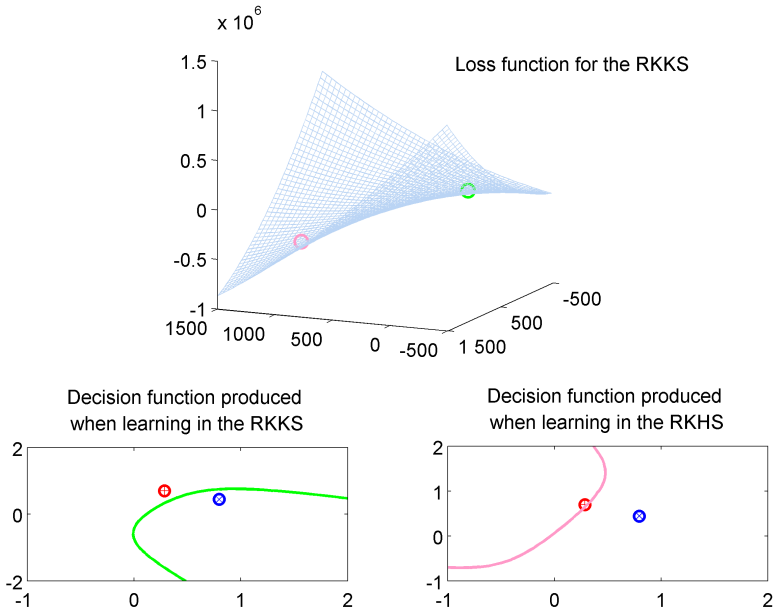

Figure 1: The above figure illustrates the effects of trying to minimize function $J(f)$ instead of stabilizing it when the kernel lies in a Kreĭn space. On the top figure, the quadratic function is represented (using the tanh kernel for 2 training points). The pink circle shows the solution provided by a standard CSVM (LibSVM). The green one shows the solution using ESVM, *i.e.* the stabilization SVM. The two figures below show the decision boundary for each of the solution (the left one corresponds to ESVM, the green circle, the right one to the CSVM, the pink circle).

minimizing. The stabilization solution is given by the ESVM algorithm proposed in this paper. The minimization solution is produced by LibSVM (Chang and Lin, 2011). For each solution, we represent its position on the loss function and the decision boundary. We can observe that the solution stabilizing the loss function gives a better decision function.

3.4 Generalization error bound

Since positive semidefiniteness is often considered by practitioners to be a "theoretical" construct, it is often assumed that this requirement comes from learning theory. However, this is not the case since a generalization error bound can be proven for indefinite kernels. We briefly reiterate this result

here to rectify this common misconception and show that positive semidefiniteness is not required to prove a generalization error bound.

This result is unsurprising since Kreĭn spaces decompose into two Hilbert spaces, and hence the corresponding Rademacher average, $R_\ell(\mathcal{F})$ for ℓ examples, can be computed for the function class $\mathcal{F}$ corresponding to an indefinite kernel.

Proposition 3.1 *(Ong, 2005, Proposition 29)*
Let $\tilde{K}$ be the Gram matrix of the kernel $\tilde{k} = k_+ + k_-$ at points $x_1, \ldots, x_\ell$. If $\tilde{k}$ is in the L_1 ball, then the Rademacher average is bounded, that is

$$R_\ell(\mathcal{B}_k) \leqslant M^{\frac{1}{2}}$$

where

$$\mathcal{B}_k = \{f \in \mathcal{K} : \|f_+\|^2 + \|f_-\|^2 = \|f\|^2 \leqslant 1\}$$

and

$$M = \int_{\mathcal{X}^d} \tilde{k}(x,x) d\mu(x).$$

The proof follows that of the Hilbertian case (Mendelson, 2003, Theorem 16) closely. Using this estimate of the complexity of the function class $\mathcal{F}$, we can obtain a generalization error bound by using standard techniques (Mendelson, 2003, Corollary 3).

4 The exact solution: using eigen-decomposition

To solve the stabilization problem of the Kreĭn space, one need to decompose f into f_+ and f_-. The idea is to identify the kernel's positive and negative components using spectral decomposition. Doing so, the stabilization formulation can be separated into a minimization subproblem and a maximization one. Some simple manipulations are then applied to obtain an equivalent standard quadratic program. We show that solving eq.2 is equivalent to solving a standard SVM with a modified kernel (*i.e.* the positive version of the indefinite kernel, built from the absolute values of the spectrum).

We first state the resulting algorithm, that computes the solution to the stabilization problem by solving the equivalent SVM dual minimization problem. We denote G the kernel matrix such that $G(i,j) = y_i y_j k(x_i, x_j)$.

This solver produces an exact solution for the stabilization problem. Its main weakness is that it requires to pre-compute the whole kernel matrix and to decompose it into eigenvectors/eigenvalues. The other point to mention

Algorithm 1 SVM solver for indefinite kernels using Eigen-decomposition (ESVM)

Require: x,y, C and G
[U,D] = EigenDecomposition(G)
$\tilde{G} = USDU^\top$ with S=sign(D)
$[\tilde{\alpha}$,b] = SvmSolver(x,y,$\tilde{G}$,C)
$\alpha = USU^\top\tilde{\alpha}$
return α,b

is that the solution α is not sparse. It can be seen as a generalization of the semi-definite case, in the sense that filling it with a positive definite kernel will produce the standard SVM solution.

Its main advantage is its simplicity, it will work with any SVM solver, and it can easily be extended to other kind of tasks or methods. Furthermore to reduce computation time, we can use partial decomposition and take only the largest eigenvalues (and associated eigenvectors) such that we keep more than, for instance, 95% of the kernel information (Williams and Seeger, 2000; Bach and Jordan, 2002; Bengio et al., 2004; Drineas and Mahoney, 2005).

In the rest of this section, we will show that the stabilization problem (Equation (2)) is indeed solved by Algorithm 1. There are four optimization problems that we consider in this section:

1. Primal stabilization problem (Equation (2))
2. Primal minimization problem (Equation (14))
3. Dual maximization problem (Equation (15))
4. Dual stabilization problem (Equation (16))

4.1 Equivalence between Stabilization and Minimization

To show the validity of Algorithm 1, we prove that the dual SVM maximization problem with an appropriately converted kernel matrix is equivalent to the primal stabilization problem. We obtain this by considering the decomposition of Kreĭn spaces into Hilbert spaces, resulting in a standard convex minimization. This standard problem allows us to use convex duality to obtain the equivalent dual maximization problem. For completeness, we derive the corresponding dual stabilization problem. We first write eq.2 according to f_+ and f_-, admitting that we have $f_\mathcal{K} = f_+ + f_-$ and $\langle f, f\rangle_\mathcal{K} = \langle f_+, f_+\rangle_{\mathcal{H}^+} - \langle f_-, f_-\rangle_{\mathcal{H}^-}$.

$$\begin{cases} \min\limits_{f_+}\max\limits_{f_-} & \frac{1}{2}\langle f_+, f_+\rangle_{\mathcal{H}^+} - \frac{1}{2}\langle f_-, f_-\rangle_{\mathcal{H}^-} \\ \min\limits_{f_+,f_-} & C\sum\limits_{i=1}^{\ell}\max\big(0, 1 - y_i(f_+(x_i) + f_-(x_i) + b)\big) \end{cases} \tag{8}$$

Note here that we replaced the stationary point search by a min-max search. We justify this decomposition below.

Proposition 4.1 *The stationary point is a saddle point.*

Here we show that the stationary point we are looking for is a saddle point, which means that we can write it as a min-max or as a max-min indifferently.

Proof 4.1.1 *To show our point, we follow Hassibi et al. (1999), section 6.3.1. Let consider the quadratic cost function*

$$J(a,b) = \begin{bmatrix} a^\top & b^\top \end{bmatrix} \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \tag{9}$$

where a and b are vectors and A,B,C are given matrices with A and C symmetric. If the middle matrix is indefinite, the solution of this quadratic form is a stationary point. Let say that one want to minimize $J(a,b)$ through the choice of a and maximize if through the choice of b, then there are two different strategies that can be applied: either the max-min problem ($\max_b \min_a J(a,b)$) or the min-max problem ($\min_a \max_b J(a,b)$). As stated in Hassibi et al. (1999) eq. 6.3.9, the condition that the min-max and max-min solutions exist simultaneously is called the saddle point condition which is A is definite positive et C is definite negative.

Now we show that our cost function $J1(f_+, f_-)$ can be written such that it is possible to identify A and C and deduce that the stationary point is a saddle point.

$$J1(f_+, f_-) = \frac{1}{2}\langle f_+, f_+\rangle_{\mathcal{H}^+} - \frac{1}{2}\langle f_-, f_-\rangle_{\mathcal{H}^-} \tag{10}$$

From eq. 6 we have $f(.) = f_+(.) - f_-(.) = C\sum_i \beta_i y_i (k_+(x_i,.) - k_-(x_i,.))$, it follows $f_+(.) = C\sum_i \beta_i y_i k_+(x_i,.)$ and $f_-(.) = C\sum_i \beta_i y_i k_-(x_i,.)$.

$$J1(\beta) = \frac{C^2}{2}\beta^\top G_+ \beta - \frac{C^2}{2}\beta^\top G_- \beta \tag{11}$$

with $G_+(i,j) = y_i y_j k_+(x_i, x_j)$ and $G_-(i,j) = y_i y_j k_-(x_i, x_j)$, so $G = G_+ - G_-$. Let use the eigen-decomposition of the indefinite kernel matrix $G =$

$UDU^\top$ where U is the orthonormal column eigenvector matrix and D the diagonal eigenvalue matrix. Since G is indefinite, D contains both positive and negative eigenvalues. Let note D_+ (resp. D_-) the diagonal submatrix of D such that it contains all and only positive (resp. negative) eigenvalues, and U_+ and U_- the submatrices of U consisting of the corresponding eigenvectors. Then $G_+ = U_+ D_+ U_+^\top$ and $G_- = U_- D_- U_-^\top$. Moreover, we denote $a = U^\top \beta = [b_+; b_-] = [U_-^\top \beta; U_+^\top \beta]$.

$$\begin{aligned} J1(\beta) = &\quad \frac{C^2}{2}\beta^\top U_+ D_+ U_+^\top \beta - \frac{C^2}{2}\beta^\top U_- D_- U_-^\top \beta \\ J1(b_+, b_-) = &\quad \frac{C^2}{2} b_+^\top D_+ b_+ - \frac{C^2}{2} b_-^\top D_- b_- \\ J1(b_+, b_-) = &\quad \begin{bmatrix} b_+^\top & b_-^\top \end{bmatrix} \begin{bmatrix} D_+ & 0 \\ 0 & D_- \end{bmatrix} \begin{bmatrix} b_+ \\ b_- \end{bmatrix} \end{aligned} \tag{12}$$

From this we can identify with eq.9 and easily see $A = D_+$ is definite positive and $C = D_-$ is definite negative. This shows that the stationary point of our problem is a saddle point and we can write it either as a min-max or a max-min system. This justifies eq.8.

The equivalent minimization system From eq.8, is it possible to change the maximization part into a minimization, and then to gather everything as follows:

$$\min_{f_+, f_-} \frac{1}{2}\langle f_+, f_+\rangle_{\mathcal{H}^+} + \frac{1}{2}\langle f_-, f_-\rangle_{\mathcal{H}^-} + C\sum_{i=1}^{\ell} \max\big(0, 1 - y_i(f_+(x_i) + f_-(x_i) + b)\big) \tag{13}$$

To establish the final minimization system, one need to note that from f_+ and f_-, we can build a positive Hilbert space, noted $\tilde{\mathcal{K}}$ such as

$$\tilde{f} = f_+ + f_- \quad \text{and} \quad \langle \tilde{f}, \tilde{f}\rangle_{\tilde{\mathcal{K}}} = \langle f_+, f_+\rangle_{\mathcal{H}^+} + \langle f_-, f_-\rangle_{\mathcal{H}^-}$$

It follows

$$\min_{\tilde{f}} \frac{1}{2}\langle \tilde{f}, \tilde{f}\rangle_{\tilde{\mathcal{K}}} + C\sum_{i=1}^{\ell} \max\big(0, 1 - y_i(\tilde{f}(x_i) + b)\big) \tag{14}$$

which is a standard SVM formulation.

4.2 Dual Optimization Problem

By using standard methods of Lagrange duality, the dual optimization problem corresponding to Equation (14) is given by

$$\begin{cases} \max\limits_{\tilde{\alpha}} & -\frac{1}{2}\tilde{\alpha}^\top \tilde{G}\tilde{\alpha} + \tilde{\alpha}^\top \mathbf{1} \\ \text{subject to} & \tilde{\alpha}^\top \mathbf{y} = 0 \\ \text{and} & 0 \leq \tilde{\alpha}_i \leq C \qquad \forall i \in [1..\ell] \end{cases} \tag{15}$$

where $\tilde{G} = G_+ + G_-$.

4.3 An equivalent stabilization problem in its dual form

The claim here is that the following stabilization system is equivalent to the primal stabilization system and is its dual form. It is shown then that the optimality conditions are the same.

The basic underlying idea is that the solution combines a positive and a negative influence. Hence the multipliers α are defined as the sum of those two components.

$$\begin{cases} \operatorname{stab}\limits_{\alpha} & -\frac{1}{2}\alpha^\top G\alpha + \alpha^\top \mathbf{1} \\ \text{subject to} & \alpha_+^\top \mathbf{y} = \alpha_-^\top \mathbf{y} \\ \text{and} & 0 \leq \alpha_{+i} \leq Cp \qquad \forall i \in [1..\ell] \\ \text{and} & -Cp \leq \alpha_{-i} \leq 0 \qquad \forall i \in [1..\ell] \end{cases} \tag{16}$$

with $\alpha = \alpha_+ + \alpha_-$

Definition 4.1 (Fundamental decomposition of $\mathcal{K}$) *(Definition 2.2.1, remarks Hassibi et al., 1999)*

We define two projection operators $\mathcal{P}_+$ and $\mathcal{P}_-$ such that

$$\mathcal{P}_+\mathcal{K} = \mathcal{K}_+ \quad \textit{and} \quad \mathcal{P}_-\mathcal{K} = \mathcal{K}_-$$

So for every $x \in \mathcal{K}$ we can write

$$x = x_+ + x_-, \quad \textit{where} \quad x_+ = P_+x \in \mathcal{K}_+ \quad \textit{and} \quad x_- = P_-x \in \mathcal{K}_-.$$

Proposition 4.2 *Matrices P_+ and P_- are given by the eigen-decomposition of the matrix G.*

Proof 4.2.1 *Using the same eigen-decomposition as for eq.12,* $G = UDU^\top$, *we can write*

$$G = U_+ D_+ U_+^\top + U_- D_- U_-^\top = G_+ - G_-$$

$$G_+ = U_+ D_+ U_+^\top = U\tilde{D}_+ U^\top \quad \textit{with} \quad \tilde{D}_+ = \begin{bmatrix} D_+ & 0 \\ 0 & 0 \end{bmatrix}$$

$$G_+ = US_+ DU^\top \quad \textit{with} \quad S_+ = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$$

$$G_+ = US_+ U^\top UDU^\top = US_+ U^\top G = U_+ U_+^\top G = P_+ G$$

The same reasoning holds for G_- *and* $P_- = -U_- U_-^\top$ *and* $G = P_+ G + P_- G$.

Then the corresponding kernel in the RKHS is written as $\tilde{G} = (P_+ - P_-)G$. We note P the projection matrix such that $P = P_+ - P_- = USU^\top$ with $S = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$.

Projection matrix between α and $\tilde{\alpha}$ Let decompose α according to P_+ and P_- and deduce the decomposition of $\tilde{\alpha}$:

$$\begin{array}{rl} \alpha = & P_+\alpha + P_-\alpha = \alpha_+ + \alpha_- \\ \tilde{\alpha} = & P_+\alpha - P_-\alpha = \alpha_+ - \alpha_- \\ = & P\alpha \end{array}$$

Note that U being orthogonal, we also have $\alpha = P\tilde{\alpha}$.

From dual maximization to dual stabilization We now use the relation between α and $\tilde{\alpha}$ in problem 3, eq.15.

$$\left\{ \begin{array}{lll} \max\limits_{\alpha} & -\frac{1}{2}\alpha^\top P\tilde{G}P\alpha + \alpha^\top P\mathbf{1} & \\ \text{subject to} & \alpha^\top P\mathbf{y} = 0 & \\ \text{and} & 0 \leq (P\alpha)_i \leq C & \forall i \in [1..\ell] \end{array} \right. \tag{17}$$

It is easy to check that $P\tilde{G}P = \tilde{G}$. Moreover, we decompose α and note that $P_+P = P_+$ and $P_-P = P_-$. We also use $P_+GP_+ = G_+$ and $P_-GP_- = G_-$:

$$\left\{ \begin{array}{lll} \max\limits_{\alpha_+,\alpha_-} & -\frac{1}{2}\alpha_+^\top G_+\alpha_+ - \frac{1}{2}\alpha_-^\top G_-\alpha_- + \alpha_+^\top\mathbf{1} - \alpha_-^\top\mathbf{1} & \\ \text{subject to} & (\alpha_+ - \alpha_-)^\top\mathbf{y} = 0 & \\ \text{and} & 0 \leq (\alpha_+ - \alpha_-)_i \leq C & \forall i \in [1..\ell] \end{array} \right. \tag{18}$$

The next step consists in changing the system such that we maximize according to α_+ and minimize according to α_-:

$$\left\{\begin{array}{lll} \max\limits_{\alpha_+}\min\limits_{\alpha_-} & -\frac{1}{2}\alpha_+^\top G_+\alpha_+ + \frac{1}{2}\alpha_-^\top G_-\alpha_- + \alpha_+^\top \mathbf{1} + \alpha_-^\top \mathbf{1} & \\ \text{subject to} & \alpha_+^\top \mathbf{y} = \alpha_-^\top \mathbf{y} & \\ \text{and} & 0 \leq \alpha_{+i} \leq C/2 & \forall i \in [1..\ell] \\ \text{and} & -C/2 \leq \alpha_{-i} \leq 0 & \forall i \in [1..\ell] \end{array}\right. \tag{19}$$

As previously, one can show that the stationary point is a saddle point, so we finally write the system as a stabilization:

$$\left\{\begin{array}{lll} \text{stab}\limits_{\alpha} & -\frac{1}{2}\alpha^\top G\alpha + \alpha\mathbf{1} & \\ \text{subject to} & \alpha_+^\top \mathbf{y} = \alpha_-^\top \mathbf{y} & \\ \text{and} & 0 \leq \alpha_{+i} \leq C/2 & \forall i \in [1..\ell] \\ \text{and} & -C/2 \leq \alpha_{-i} \leq 0 & \forall i \in [1..\ell] \end{array}\right. \tag{20}$$

The three previous subsections have shown that the four considered problems are equivalent and hence Algorithm 1 solves the proposed SVM like stabilization algorithm (Equation (2)).

5 Approximate solutions: avoiding the spectral decomposition

For large scale problems, the computation cost of the eigendecomposition proposed in the previous section may be prohibitive. In this section, the idea is to find an approximate solution to the non positive SVM that will not require the storage of the full kernel matrix and lead to a sparse solution. Instead of modifying the kernel matrix, we directly adapt the quadratic program.

5.1 Interpretation of Karush Kuhn Tucker conditions

We begin from the SVM dual quadratic problem (Vapnik, 1995), which is obtained from the margin maximization formulation. A positive Lagrange multiplier α_i is associated to each training example.

$$\left\{\begin{array}{lll} \min\limits_{\alpha} & \frac{1}{2}\alpha^\top G\alpha - \alpha^\top \mathbf{1} & \\ \text{st} & \alpha^\top \mathbf{y} = 0 & \\ & 0 \leq \alpha_i \leq C & \forall i \in [1..n] \end{array}\right. \tag{21}$$

where $G(i,j) = y_i y_j K(i,j)$. The full Karush Kuhn Tucker (KKT) conditions for the classical SVM dual system (21) are as follows:

$$\begin{array}{lcll} \text{Stationarity} & \alpha^\top G - \mathbf{1}^\top + \lambda \mathbf{y}^\top - \mu^\top + \eta^\top = \mathbf{0} & & \\ \text{Primal admissibility} & \alpha^\top \mathbf{y} = 0 & 0 \leq \alpha_i \leq C & \forall i \in [1..\ell] \\ \text{Dual admissibility} & \eta_i \geq 0 & \mu_i \geq 0 & \forall i \in [1..\ell] \\ \text{Complementarity} & \lambda \alpha^\top \mathbf{y} = 0 & \mu_i \alpha_i = 0 & \eta_i(\alpha_i - C) = 0 \end{array} \tag{22}$$

Any solution respecting each of these conditions is a solution of the problem 21. In the case the kernel matrix is definite positive, the dual SVM problem has a unique solution which is a global minimum. When the kernel matrix is indefinite, we directly consider the KKT conditions, which has three possible interpretations which will be discussed in the following sections.

5.1.1 Point of view 1: the variational approach of quadratic programming

In the case the kernel matrix is indefinite, the dual SVM problem is not well defined and the solution is not unique. Following Brezinski (1997) and the variational approach of quadratic programming, the problem can actually be solved using normal residuals (ie. solving $Ax = b$ via $A^\top Ax = A^\top b$). Note that normal equations could also be an option (ie. solving $Ax = b$ via $AA^\top x' = b$ with $x = A^\top x'$). Applying these variational approaches to the stationarity condition results in:

$$\begin{array}{rcl} \alpha^\top G & = & \mathbf{1} - \lambda \mathbf{y}^\top + \mu^\top - \eta^\top \\ \alpha^\top G G^\top & = & (\mathbf{1} - \lambda \mathbf{y}^\top + \mu^\top - \eta^\top) G^\top \end{array} \tag{23}$$

with all the other KKT conditions remaining identical. This approach can be interpreted as solving the linear equation with a least squares method.

5.1.2 Point of view 2: a stabilization problem

Optimization techniques provide minima and not saddle points. To that purpose, a simple trick is proposed, known as magnitude of the gradient, consisting in changing the problem such that any critical point becomes a local minimum. This is done by computing the sum of the squares of the partial derivatives of the function to be stabilized. We apply this to the following unconstrained function:

$$\mathcal{J} = \frac{1}{2} \alpha^\top G \alpha - \alpha^\top \mathbf{1} \tag{24}$$

Stabilizing $\mathcal{J}$ is equivalent to minimizing $\mathcal{M}$:

$$\mathcal{M}(\alpha) = \left\langle \alpha^\top G - \mathbf{1}^\top, \alpha^\top G - \mathbf{1}^\top \right\rangle \tag{25}$$

This provides the following system:

$$\begin{cases} \min\limits_{\alpha} & \left\langle \alpha^\top G - \mathbf{1}^\top, \alpha^\top G - \mathbf{1}^\top \right\rangle \\ \text{st} & \alpha^\top \mathbf{y} = 0 \\ & 0 \leq \alpha_i \leq C \qquad\qquad \forall i \in [1..n] \end{cases} \tag{26}$$

The KKT conditions of optimality for system (26) gives following stationarity condition:

$$(\alpha^\top G - \mathbf{1}^\top + \lambda y^\top - \mu^\top + \eta^\top) G^\top = \mathbf{0} \tag{27}$$

The other conditions remain identical to (22).

5.1.3 Point of view 3: The projection

As already mentioned, the unconstrained problem has a unique solution. The objective is to project it onto the feasible set. However, this projection is not obvious, namely what is the closest point of the feasible set to the unconstrained optimum in the sense of the stabilization? We propose to define it as the most stable point, *i.e.* the admissible point minimizing the gradient of the cost function (which is $\alpha^\top G - \mathbf{1}^\top$). Solving this minimization with the least squares directly gives the same system (26).

5.2 Approximate solver for stabilization

The proposed algorithm is derived from active set approach for SVM, similar to Vishwanathan et al. (2003). The sets of points are defined according to the complementarity conditions (see Table 1). We initialize all training points are in the non support vector set I_0 except for a couple with opposite labels which is in I_w. Any other initial situation based on warm-start or a priori does not change the algorithm.

Table 1: Definition of groups for active set depending on the dual variable values

Group	α	η	μ
I_0	0	0	> 0
I_C	C	> 0	0
I_w	$0 < \alpha < C$	0	0

Solving linear system in I_w The linear system is solved from the stationarity condition (eq. 27) only for unconstrained points, those of I_w. This leads to the following equation:

$$\alpha_{(w)}^\top G_{(w,:)} G_{(:,w)} = (\mathbf{1}_{(:)} - \lambda \mathbf{y}_{(:)}^\top - C\mathbf{1}_{(C)}^\top G_{(C,:)}) G_{(:,w)} \tag{28}$$

This can be solved using QR decomposition of G, for which one can maintain a rank one update at each step of the algorithm. Computing λ can be easily done substituting $\alpha_{(w)}^\top$ in $\alpha_{(w)}^\top \mathbf{y}_{(w)} = -C\mathbf{1}_{(C)}^\top \mathbf{y}_{(C)}$.

Activating constraints in I_w If any $\alpha_{(w)}(i)$ does not lie in $[0 \quad C]$, the current solution is projected on the admissible set such that all $\alpha_{(w)}(i)$ satisfy the primal admissibility and the violating point is transferred towards I_0 or I_C according to the violating value.

Relaxing constraints in I_0 or I_C If the current solution is admissible, the stationarity conditions for I_0 and I_C (using eq. 27) is checked. The most violating point is transferred from its group to I_w.
For any point $j \in I_0$, $\mu_j > 0$ and $\eta_j = 0$. From eq. 27 :

$$(\alpha_{[w,C]}^\top G_{([w,C],:)} - \mathbf{1}^\top + \lambda \mathbf{y}^\top) G_{:,j} > 0 \tag{29}$$

For any point $k \in I_C$, $\mu_k = 0$ and $\eta_k > 0$. From eq. 27 :

$$(\alpha_{[w,C]}^\top G_{([w,C],:)} - \mathbf{1}^\top + \lambda \mathbf{y}^\top) G_{:,k} < 0 \tag{30}$$

The notion of margin is distorted. Indeed, when using the same active set solver in the positive definite case, the margin clearly appears in the constraint relaxation (for $j \in I_0$, the condition would be $\alpha_{[w,C]}^\top G_{([w,C],j)} + \lambda y_j > 1$). This means that in the feature space, the solution will not have the same properties as the usual SVM, especially concerning the interpretation of support vectors relative to the decision boundary.

Note that the proposed algorithm converges after a finite number of steps since it can be seen as an active set procedure applied to a convex QP and thus convergence proof in this case applies (Nocedal and Wright, 2006).

Complexity The proposed algorithm (Algorithm 2) is slightly more computationally intensive than the positive semidefinite algorithm. By assuming that the original kernel matrix is stored in memory, we have the following additional operations

Algorithm 2 Kreĭn space SVM solver (KSVM)

Initialize (one random point for each class in I_w, all others in I_0)
while solution is not optimal **do**
  solve linear system (28).
  if primal admissibility is not satisfied **then**
    project solution in the admissible domain:
    remove a support vector from I_w (to I_0 or I_C).
  else if stationarity condition is not satisfied **then**
    add new support vector to I_w (from I_0 (29) or I_C (30)).
  end if
end while

- 2 matrix by matrix multiplications ($\mathcal{O}(n|I_w|^2)$ and $\mathcal{O}(n^2|I_w|)$) to solve Equation (28).
- Activating the constraints is identical to the positive semidefinite case.
- 1 matrix by matrix multiplication ($\mathcal{O}(n^2(|I_0|+|I_C|))$) in Equation (29) and (30).

These can be reduced by various strategies, such as caching, rank-one updates, and iterative search for violating examples Vishwanathan et al. (2003).

6 Empirical comparison

We perform empirical comparisons of our three proposed approaches to previous methods for indefinite kernels. All experiments are realized using Matlab code available at
`http://gaelle.loosli.fr/npsvm.html`.

6.1 Overview of related work

In addition to the three approaches proposed in this paper, Algorithm 1 (ESVM) with full and partial eigenvalue decomposition, and Algorithm 2 (KSVM) there are several recent approaches for learning with indefinite kernels.

6.1.1 Transforming the kernel matrix

There are several common ways of converting indefinite kernel matrices to positive semidefinite ones by changing the eigenspectrum (Muñoz and

de Diego, 2006).

Absolute values: In this method, the kernel matrix is decomposed in eigenvectors and eigenvalues and recomposed using the absolute values of the eigenvalues. Hence large negative components become large positive components.

Truncating values: In this method, the kernel matrix is decomposed in eigenvectors and eigenvalues and recomposed using only the positive eigenvalues, the negative part is set to zero. Hence all negative components are removed.

Shifting values: In this method, the kernel matrix is decomposed in eigenvectors and eigenvalues and recomposed using a translated spectrum: the largest negative eigenvalue is subtracted to all the others. Hence large negative components become the smallest positive components.

6.1.2 Direct computation with indefinite kernels

We also compared with two other methods that perform classification with indefinite kernels:

IndefiniteSVM: The proposed method can be seen as a penalized kernel learning problem where indefinite kernel matrices are treated as noisy observations of a true Mercer kernel. The IndefiniteSVM is successfully compared to other de-noising approaches (Luss and d'Aspremont, 2009). The Matlab implementation used for the experiments can be found on the author's webpage [1].

Relevance Vector Machine (RVM): This technique uses Bayesian inference to obtain parsimonious solutions that have an identical functional form to the SVM, but provides probabilistic classification. Moreover it handles indefinite kernels directly, so RVM is a good challenger for ESVM and KSVM (Tipping (2001)). The implementation used for the experiments is the SB2 release200 [2].

LP SVM: Linear Programming SVM, following Mangasarian et al. (1999), in which the formulation does not require the kernel to be positive. The implementation is based on the Matlab's `linprog`.

[1] `http://www.eecs.berkeley.edu/~rluss/`

[2] `http://www.vectoranomaly.com/downloads/downloads.htm`

6.1.3 Computing the kernel on the test set

Algorithm 1 shows that ESVM and ESVM-L use a usual SVM solver, after applying the absolute values trick explained in section 6.1.1. It is important to underline the practical difference of this trick with the proposed approach: the classical solver provides a solution that lies in a RKHS space for which the kernel formulation is unknown: only the transformed kernel for training points is available. This implies that testing a new point, using the known kernel of the RKKS space may produce unwanted results (and this explains the very poor results obtained by the heuristics in the experimental part since all performance rates are given using the original kernel). The most common approach to deal with this obvious drawback is to transform the kernel including the test data. More elegantly, Gu and Guo (2012) propose a method jointly compute the SVM solution and the transformation of the matrix such that it can be applied to the mixed kernel matrix between support vectors and test data. On the contrary, ESVM and ESVM-L transform the RKHS solution into a RKKS solution via those simple operations:

```
[U,D] = eig(G);           % G is the Krein space kernel
S = sign(D);              % S identifies the negative eigenvalues
K = U*S*D*U';             % K is the  Hilbert space kernel
[alphat,b] = SvmSolver(x,y,K,C);     % solution in the RKHS
alpha = U*S*U'*alphat;               % solution in the RKKS
```

Doing that, the final solutions of ESVM and ESVM-L can be tested for any new point since it is based on the true kernel. KSVM do not transform the kernel so the solution naturally uses the true kernel.

6.2 Checkerboard datasets

We illustrate the difference between the different approaches on a synthetic 2D dataset where the datapoints are labelled in a checkerboard pattern.

6.2.1 ESVM with partial decomposition

As a sanity check, Figure 2 illustrates that performing a partial eigendecomposition of the kernel matrix does not harm the classification accuracy of ESVM. This is not surprising since most of the kernel information is captured by the eigenvectors corresponding to the first largest eigenvalues. Moreover, keeping only the largest eigenvalues/eigenvectors often cleans the data and provide better results than the exact decomposition.

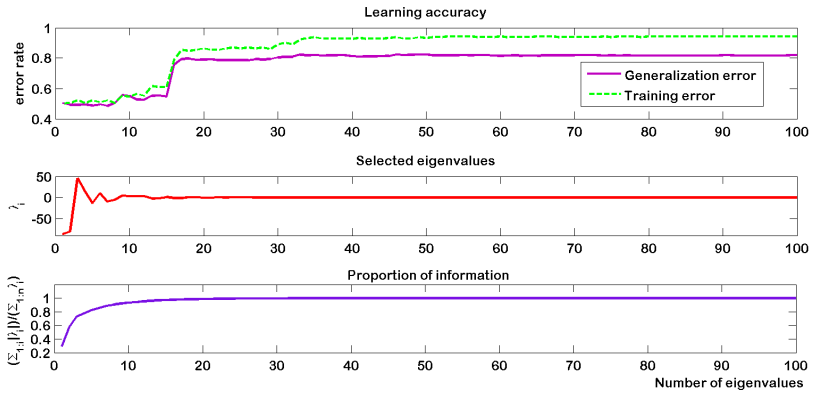

Figure 2: The top graph shows the training and generalization error on a noisy checkerboard problem, depending on the number of eigenvectors/eigenvalues kept for decomposition. The middle and bottom graphs represent respectively the eigenvalues (sorted by absolute value) and their cumulated sum (represented as the proportion of total sum of eigenvalues). We can observe that the error rates are stable once all the meaningful eigenvalues are taken into account for the kernel decomposition.

6.2.2 Comparison to other methods

We compare the algorithms discussed in Section 6.1 on the synthetic data to show its behavior and scalability.

Quick illustration of each method On Figures 3 and 4, the results of each method are shown, on a separable checkerboard dataset, trained on 500 and 180 points respectively. Results on Figure 3 illustrates the danger of forcing the kernel to be positive semidefinite. The methods in the second row show poor estimation of the classification boundaries due to the loss of information in the negative eigenspectrum. For this experiment, the sigmoid kernel is used, its largest negative eigenvalue is -231.77.

Results on Figure 4 illustrate that our proposed methods (ESVM, ESVM-L and KSVM) are more reliable than RVM when fewer training points are available. For this experiment, the sigmoid kernel is used, its largest negative eigenvalue is -187.46.

Computational time In order to evaluate the computational complexity of the different methods for non positive SVM, the following experiment

has been carried out: the training time of each method is computed for increasing training sizes of the binary checkerboard with 10% of overlapping classes at boundaries. The experiment is run 10 times and the curves of Figure 5 are the average training time. We observe that KSVM and RVM have a similar experimental complexity, while ESVM suffers from the complete eigen-decomposition. ESVM-L is the most competitive method as far as training time is concerned, due to the partial eigen-decomposition. IndefiniteSVM and LP SVM are the most complex methods to train. All software are fully in Matlab, RVM and IndefiniteSVM are provided by their respective authors.

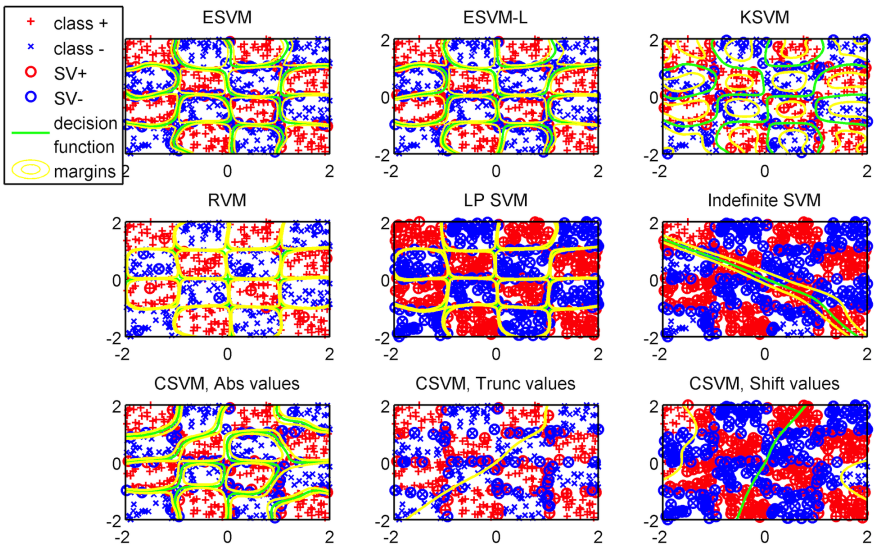

Figure 3: This set of experiments show the results on a checkerboard dataset for several algorithms dealing with non positive kernel matrices. On the first row are shown ESVM, ESVM-L, KSVM . On the second row are RVM, LP SVM and IndefiniteSVM (with projected gradient). On the third row are C-SVM with 3 different tricks to transform the matrix into a definite-positive matrix: taking the absolute values of the eigenvalues, cutting all eigenvalues below zero, or translating the all spectrum into the positive space. This experiment is trained on 500 points. The methods that force positive semi-definiteness show poor estimation of the classification boundary, except for absolute values.

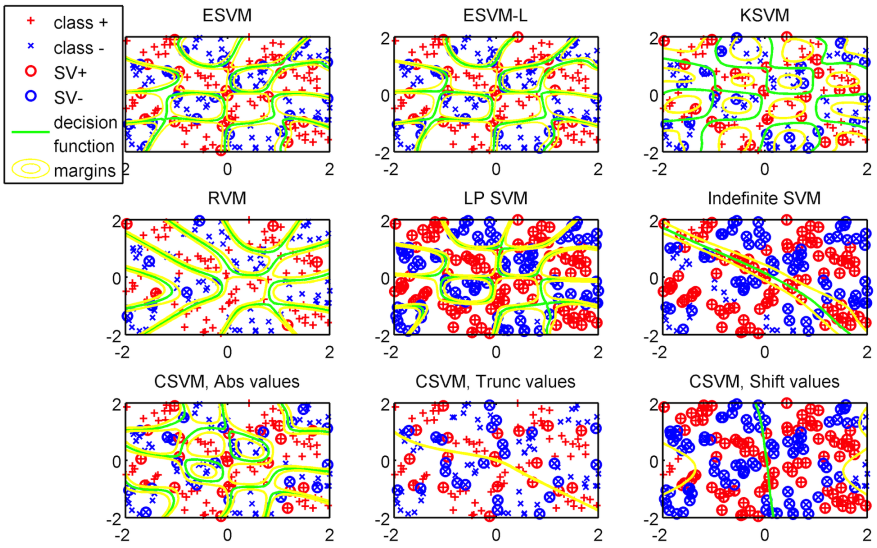

Figure 4: Training with 180 examples. The proposed methods (ESVM, SVM-L and KSVM) are more reliable than RVM when fewer training points are available. See Figure 3 for the description of the subfigures.

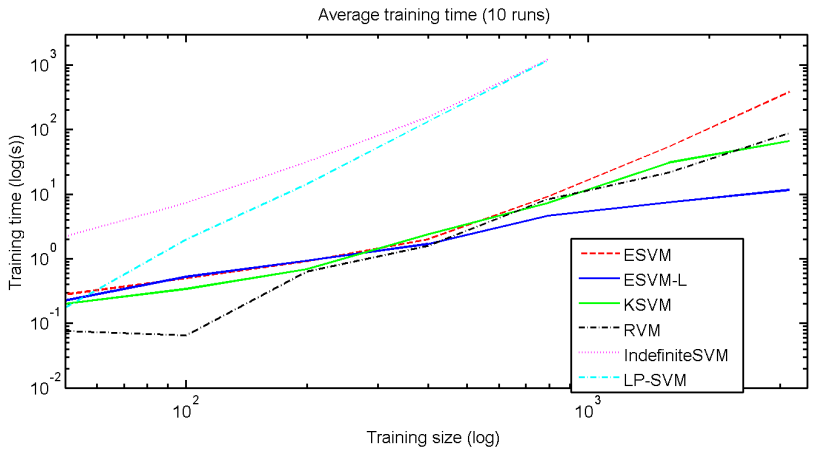

Figure 5: This set of experiments shows the relative training time for different methods, for a increasing size problem (a binary checkerboard, with 10 % of overlapping classes). Those curves are obtained on average, for 10 runs.

Table 2: Results on various UCI datasets, for different settings of the sigmoid kernel. Each line gives the 5-fold CV performance (%) of each learning method, and the standard deviation on 10 runs in brackets. The least eigenvalue of the training kernels is given after the dataset name. ESVM-L is computed keeping 90% of the eigenvalues information. The best performance is shown in bold. For all experiments, the best performance is consistently given either by one the proposed method of the paper, LP SVM or RVM.

Problem (min eig)	**ESVM**	**ESVM-L**	**KSVM**	**RVM**	**Indefinite SVM**	**LP SVM**	**Abs eigs**	**Trunc eigs**	**Shift eigs**
Sonar	71.42	71.42	69.60	70.20	50.51	**73.17**	53.87	46.12	51.84
(-184.13)	(1.46)	(1.46)	(3.56)	(3.79)	(0.46)	**(2.35)**	(1.25)	(4.75)	(3.08)
Heart	**82.74**	**82.74**	70.14	80.96	61.11	81.92	22.96	55.55	20.22
(-148.25)	**(2.67)**	**(2.67)**	(5.66)	(1.72)	(1.07)	(0.71)	(1.11)	(0.37)	(0.42)
Breast cancer	**97.29**	**97.29**	96.66	95.52	66.14	91.51	87.15	30.33	78.59
(-548.24)	**(0.15)**	**(0.15)**	(0.23)	(0.62)	(2.99)	(11.44)	(0.10)	(2.60)	(3.76)
Diabetes	75.10	75.10	72.03	**76.66**	36.19	76.25	29.60	27.19	64.74
(-235.64)	(1.93)	(1.93)	(2.45)	**(0.62)**	(0.56)	(0.42)	(3.64)	(2.37)	(0.13)
Adult (a1a)	**80.66**	**80.66**	79.25	79.66	75.38	**80.70**	50.92	35.17	74.65
(-1452.8)	**(0.62)**	**(0.62)**	(0.60)	(0.63)	(0)	**(0.48)**	(1.14)	(0.64)	(0.53)

6.3 Comparisons on benchmark datasets

6.3.1 UCI Machine Learning Repository

This experiment is done on datasets from UCI (A. Asuncion, 2007). Several indefinite sigmoid kernels are built and used with each method. Some results, based on 5-fold cross validation are provided in Table 2. The goal here is not to find the best kernel for this dataset, but rather to visualize the general behavior of each approach. Figures 6 and 7 in the appendix show the complete results for Heart and Sonar datasets, for all methods, depending on the least eigenvalue of the training kernel matrix. On each figure, the same general behavior can be observed: ESVM, LP-SVM and RVM are almost always not only better than the others but also good in accuracy. IndefiniteSVM is regularly better than the heuristics (but not necessarily accurate). The heuristics perform badly most often, except when the least eigenvalue is quite small in absolute value.

6.3.2 Dissimilarity dataset

In this part, experiments on dissimilarity dataset are shown. Here, we use a linear kernel and train the SVM on the dissimilarity measures. Datasets are

provided by the Pattern Recognition Lab of Delft University of Technology [3]. We used 8 of the proposed dataset: Balls3D (200 points, 2 classes), GaussM1 (2000 points, 2 classes), GaussM02 (2000 points, 2 classes), CatCortex (65 points, 4 classes), Protein (213 points, 3 classes), CoilYork (288 points, 4 classes), Newgroups (600 points, 4 classes), PolyDisH57 (4000 points, 2 classes). For multiclass problems, we arbitrarily transform them into binary problems. Note that as previously, the performance is computed using the original kernel and not the modified one, to simulate what would happen if new data were presented.

Whereas for some datasets in UCI, some heuristics can do as well as the native non positive methods, we can see from Table 3 that this does not happen with these datasets. Our proposed methods perform best in all the datasets except for GaussM02 where we are a close second.

7 Conclusion

This paper aims at providing computational methods for solving a binary classification problem with an indefinite kernel. More generally, the underlying claim is that the negative part of an indefinite kernel cannot be systematically considered as noise. Indeed, the negative part of an indefinite kernel can carry some important information. This point of view discards all methods that tend to *correct* an indefinite kernel, and a striking example is provided in the experimental part: a simple problem (checkerboard) with a simple kernel (sigmoid) exhibits the loss of information induced by any correction of the kernel (section 6.2.2).

When dealing with an indefinite kernel, one has to solve the learning problem in an RKKS. This implies the resolution of a stabilization system under constraints. This setting lacks theoretical tools to be tackled. In this paper, the equivalence between a *primal* stabilization problem and a *dual* stabilization problem is shown, based on the optimality conditions. To achieve this, the primal stabilization system is written under a minimization system, using the fact that the kernel of a Kreĭn space is the difference of two kernels lying in some Hilbert spaces. Taking the dual of the minimization system is a well known task. The equivalence between the proposed *dual* stabilization system and the dual maximization system is shown. This last part of the reasoning directly provides a direct algorithm based on the eigendecomposition of the kernel matrix. As shown in the experimental part, ESVM outperforms RVM, which also deals with indefinite kernels and it

[3] `http://prtools.org/disdatasets/`

Table 3: Results on various DDPR datasets, for linear kernels. Each line gives the 5-fold CV performance (%) of each learning method, and the variance on 10 runs. The least eigenvalue of the training kernels is given under the dataset name. The best performance is shown in bold. The C value is found by cross validation. ESVM-L is computed keeping 90% of the eigenvalues information. For all experiments, the best performance is consistently given either by one the proposed method of the paper, LP-SVM or RVM. Missing results are due to a lack of convergence or prohibitive training time.

Problem (min eig)	**ESVM**	**ESVM-L**	**KSVM**	**RVM**	**Indefinite SVM**	**LP SVM**	**Abs eigs**	**Trunc eigs**	**Shift eigs**
Balls3D	**53.90**	**53.90**	**53.98**	47.40	49.99	53.30	45.89	45.10	44.41
(-2.8894e+03)	**(2.47)**	**(2.47)**	**(1.58)**	(0.82)	(0.01)	(3.86)	(2.40)	(1.54)	(2.07)
GaussM1	81.94	81.94	**83.95**	82.40	-	-	17.67	16.54	16.89
linear (-1.8739e+03)	(0.43)	(0.43)	**(0.09)**	(0.37)	(-)	(-)	(0.42)	(0.92)	(0.21)
GaussM02	73.98	73.98	78.75	**79.54**	-	-	23.42	38.31	16.24
(-2.1096e+08)	(0.70)	(0.70)	(0.84)	**(0.51)**	(-)	(-)	(0.78)	(0.76)	(0.29)
CatCortex,2vs1	**95.71**	**95.71**	63.59	90.47	55.46	81.31	8.31	26.13	10.12
(-24.61)	**(2.30)**	**(2.30)**	(5.27)	(2.96)	(0.10)	(8.12)	(1.07)	(8.71)	(2.53)
Protein,2vs2	**99.91**	**99.91**	48.54	98.21	52.11	98.59	0.09	7.98	1.87
(-322.34)	**(0.21)**	**(0.21)**	(1.43)	(0.39)	(0.00)	(1.10)	(0.21)	(6.23)	(0.66)
CoilYork,2vs2	**75.83**	**75.83**	65.00	74.16	-	70.55	24.44	30.06	31.04
(-9.4559e+03)	**(1.87)**	**(1.87)**	(2.22)	(3.40)	(-)	(6.19)	(4.38)	(1.58)	(0.67)
NewsGroups,2vs2	**87.80**	**87.80**	71.16	84.40	50.83	65.36	12.06	17.93	14.06
(-16.39)	**(0.32)**	**(0.32)**	(0.97)	(1.32)	(0.12)	(3.48)	(0.71)	(4.79)	(0.84)
PolyDisH57	**99.75**	**99.75**	63.21	98.90	-	-	8.72	15.67	18.77
(-535.96)	**(0.08)**	**(0.08)**	(0.38)	(0.18)	(-)	(-)	(0.40)	(0.77)	(0.26)

is even more visible when the number of training data is relatively small (figure 4). LP-SVM can be as accurate as the proposed method, however it comes with a lot of support vectors and a higher complexity (figures 3 and 5). While ESVM requires to decompose the kernel, which can be expansive, ESVM-L works with only a partial decomposition and it turns out that most of times, it is as accurate as ESVM. If the kernel cannot be precomputed, an other approach is proposed, based on the interpretation of the meaning of the stabilization. This leads to the KSVM algorithm. This method is less accurate than ESVM which is not surprising since it is an approximate version. As long as it is possible to handle the eigen-decomposition of the kernel matrix, ESVM (or ESVM-L) is a better choice.

The interest of dealing with non modified indefinite kernels is clearly shown in this paper. This opens a wide range of future work, including the

development of theoretical tools to deal with the stabilization setting or the extension to other kernel methods and tasks.

Acknowlegment

This work was supported in part by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886. NICTA is funded by the Australian Government's Department of Communications, Information Technology and the Arts, the Australian Research Council through Backing Australia's Ability, and the ICT Centre of Excellence programs.

A Detailed UCI results

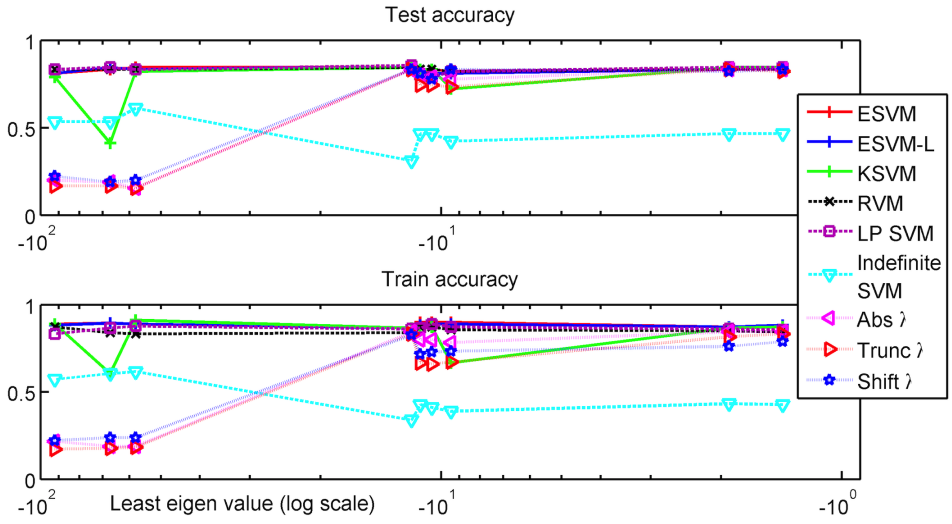

Figure 6: Results for the Heart dataset. The top figure shows the accuracy on the test set, the bottom one shows the accuracy on the training set, computed using the original kernel, even for heuristics. ESVM and RVM are comparable and perform well. KSVM has an inconstant performance. The 3 heuristics perform poorly when the kernel matrix has very large negative eigenvalues and perform quite well when negative eigenvalues are quite small. IndefiniteSVM lies in between.

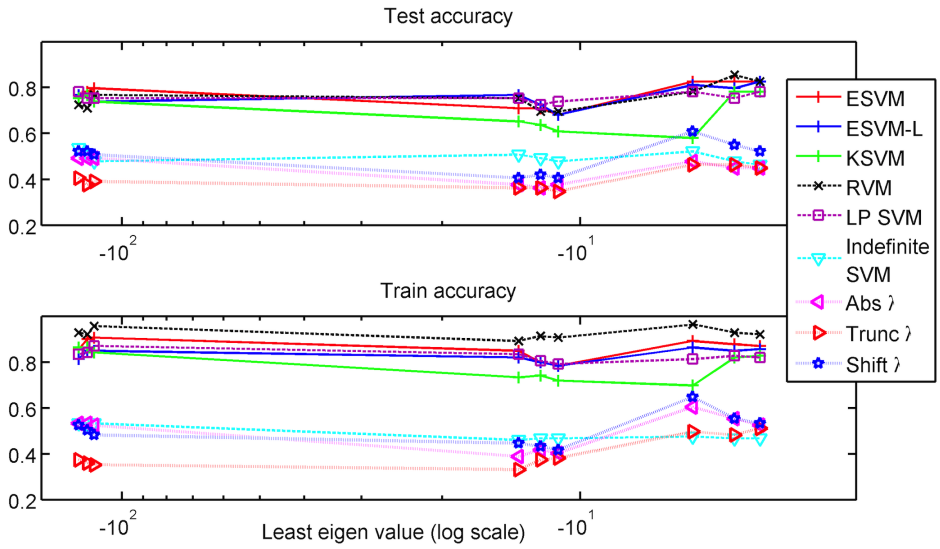

Figure 7: Results on Sonar dataset. This experiment shows roughly the behavior as the previous one, but it also illustrates that among the heuristics, the weakest one is *Trunc*, ie. the one that simply ignores the negative part.

References

D.J. Newman A. Asuncion. UCI machine learning repository, 2007. URL `http://www.ics.uci.edu/$\sim$mlearn/{MLR}epository.html`.

N.I. Akhiezer and I.M. Glazman. *Theory of Linear Operators in Hilbert Space*. Dover, 1993.

N. Aronszajn. Theory of reproducing kernels. *Trans. American Mathematical Society*, 68:337–404, 1950.

T. Ya. Azizov and I. S. Iokhvidov. *Linear Operators in Spaces with an Indefinite Metric*. Wiley, 1989. Translated by E. R. Dawson.

F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

Y. Bengio, O. Delalleau, N. Le Roux, J.-F. Paiement, P. Vincent, and M. Ouimet. Learning eigenfunctions links spectral embedding and kernel PCA. *Neural Computation*, 16(10):2197–2219, 2004.

János Bognár. *Indefinite Inner Product Spaces*. Springer, 1974.

Sabri Boughorbel, Jean-Philippe Tarel, and Francois Fleuret. Non-mercer kernels for svm object recognition. In *In British Machine Vision Conference (BMVC*, pages 137–146, 2004.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

C. Brezinski. Projection methods for linear systems. *J. Comput. Appl. Math.*, 77(1-2):35–51, 1997. ISSN 0377-0427. doi: http://dx.doi.org/10.1016/S0377-0427(96)00121-5.

C.-C. Chang and C.-J. Lin. LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(27): 1–27, 2011.

Jianhui Chen and Jieping Ye. Training svm with indefinite kernels. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *ICML*, volume 307 of *ACM International Conference Proceeding Series*, pages 136–143. ACM, 2008. ISBN 978-1-60558-205-4.

Yihua Chen, Maya R. Gupta, and Benjamin Recht. Learning kernels from indefinite similarities. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 145–152, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553393. URL http://doi.acm.org/10.1145/1553374.1553393.

F.H. Clarke. *Optimization and nonsmooth analysis*. Centre de Recherches Mathématiques, 1989. URL http://books.google.fr/books?id=pUnvAAAAMAAJ.

Michael Collins and Nigel Duffy. Convolution kernels for natural language. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *NIPS*, pages 625–632. MIT Press, 2001. URL http://books.nips.cc/papers/files/nips14/AA58.pdf.

Corinna Cortes, Patrick Haffner, and Mehryar Mohri. Positive definite rational kernels. In *In Proceedings of The 16th Annual Conference on Computational Learning Theory (COLT 2003*, pages 41–56. Springer, 2003.

Petros Drineas and Michael W. Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

Suicheng Gu and Yuhong Guo. Learning svm classifiers with indefinite kernels. In Jörg Hoffmann and Bart Selman, editors, *AAAI*. AAAI Press, 2012.

Bernard Haasdonk. Feature space interpretation of svms with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:482–492, 2005. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2005.78.

Bernard Haasdonk and Hans Burkhardt. Invariant kernel functions for pattern analysis and machine learning. In *Machine Learning*, pages 35–61, 2007.

Bernard Haasdonk and Elzbieta Pekalska. Indefinite kernel fisher discriminant. In *International Conference on Pattern Recognition*, 2008. URL http://www.ians.uni-stuttgart.de/publications/2008/HP08a.

Babak Hassibi, Ali H. Sayed, and Thomas Kailath. *Indefinite-quadratic estimation and control: a unified approach to H2 and H [infinity] theories*, volume 16. SIAM, 1999. ISBN 0898714117.

Bernd Hofmann, Peter Mathé, and Sergej V Pereverzev. Regularization by projection: Approximation theoretic aspects and distance functions. *Journal of Inverse and Ill-posed Problems jiip*, 15(5):527–545, 2007.

Lin Hsuan-Tien and Lin Chih-Jen. A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan., 2003.

Matthieu Kowalski, Marie Szafranski, and Liva Ralaivola. Multiple indefinite kernel learning with mixed norm regularization. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 545–552, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553445. URL `http://doi.acm.org/10.1145/1553374.1553445`.

Julian Laub and Klaus-Robert Müller. Feature discovery in non-metric pairwise data. *J. Mach. Learn. Res.*, 5:801–818, 2004. ISSN 1532-4435.

G. Loosli and S. Canu. Non positive svm. In *4th International Workshop on Optimization for Machine Learning,NIPS*, 2011.

R. Luss and A. d'Aspremont. Support Vector Machine Classification with Indefinite Kernels. *Mathematical Programming Computations*, 2009.

Olvi L Mangasarian et al. Generalized support vector machines. *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pages 135–146, 1999.

Shahar Mendelson. A few notes on statistical learning theory. In S. Mendelson and A. J. Smola, editors, *Advanced Lectures in Machine Learning*, LCNS2600, pages 1–40. Springer, 2003.

Alberto Muñoz and Issac Martín de Diego. From indefinite to positive semidefinite matrices. In *SSPR&SPR 2006*, LNCS 4109, pages 764–772, 2006.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, second edition, 2006. ISBN 0387987932.

Cheng Soon Ong. *Kernels: Regularization and Optimization*. PhD thesis, The Australian National University, 2005.

Cheng Soon Ong, Xavier Mary, Stéphane Canu, and Alexander J. Smola. Learning with non-positive kernels. In *ICML '04: Proceedings of the*

twenty-first international conference on Machine learning, page 81, New York, NY, USA, 2004. ACM. ISBN 1-58113-828-5. doi: http://doi.acm.org/10.1145/1015330.1015443.

Elzbieta Pekalska and Bernard Haasdonk. Kernel discriminant analysis for positive definite and indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1017–1032, 2009. URL `http://www.ncbi.nlm.nih.gov/pubmed/19372607`.

Ralph Tyrell Rockafellar. *Convex Analysis*. Princeton University Press, reprint edition, 1996.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. MIT Press, 2002.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer, 2008.

Hongwei Sun and Qiang Wu. Least square regression with indefinite kernels and coefficient regularization. *Applied and Computational Harmonic Analysis*, 30(1):96 – 109, 2011. ISSN 1063-5203. doi: 10.1016/j.acha.2010.04.001. URL `http://www.sciencedirect.com/science/article/pii/S1063520310000515`.

Michael E. Tipping. Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244, September 2001. ISSN 1532-4435. doi: 10.1162/15324430152748236. URL `http://dx.doi.org/10.1162/15324430152748236`.

Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer, N.Y, 1995.

S. V. N. Vishwanathan, Alex J. Smola, and M. Narasimha Murty. Simplesvm. In *ICML*, pages 760–767, 2003.

Christopher Williams and Matthias Seeger. The effect of the input density distribution on kernel-based classifiers. In P. Langley, editor, *International Conference on Machine Learning 17*, pages 1159–1166. Morgan Kaufmann, 2000.

Yiming Ying, Colin Campbell, and Mark Girolami. Analysis of svm with indefinite kernels. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 2205–2213, 2009.