Learning SVM in Kreĭn Spaces

Gaëlle Loosli, Stéphane Canu, and Cheng Soon Ong,

Abstract—This paper presents a theoretical foundation for an SVM solver in Kreĭn spaces. Up to now, all methods are based either on the matrix correction, or on non-convex minimization, or on feature-space embedding. Here we justify and evaluate a solution that uses the original (indefinite) similarity measure, in the original Kreĭn space. This solution is the result of a stabilization procedure. We establish the correspondence between the stabilization problem (which has to be solved) and a classical SVM based on minimization (which is easy to solve). We provide simple equations to go from one to the other (in both directions). This link between stabilization and minimization problems is the key to obtain a solution in the original Kreĭn space. Using KSVM, one can solve SVM with usually troublesome kernels (large negative eigenvalues or large numbers of negative eigenvalues). We show experiments showing that our algorithm KSVM outperforms all previously proposed approaches to deal with indefinite matrices in SVM-like kernel methods.

Index Terms—dissimilarity, Kreĭn spaces, SVM, indefinite kernel, stabilization problem, classification

1 Introduction

Training a support vector machine (SVM) with indefinite kernel matrices has been a regular subject of interest since the beginning of SVM. While theoretically not allowed, matrices that are not positive definite are used in several application fields, and some toolboxes even include specific treatments to tolerate such matrices. There are also several heuristics that are applied as pre-treatment to cure the indefinite matrices, and more recently, even dedicated methods. In this paper, we consider the signal to truly belong in indefinite space, and in contrast to related work we do not seek to distort the kernel. Taking this point of view [1] proposed a stabilization approach when faced with an indefinite kernel, instead of the standard minimization approach.

However the proposed solution was not directly applicable to the SVM, and there has been several different proposals for solving the SVM problem with indefinite kernels in the subsequent years. Unfortunately none of the proposed approaches address the theoretically desirable stabilization problem. Recently [2] proposed a solution to stabilization problem for the SVM, using the eigen-decomposition of the kernel matrix. This paper extends [2], surveying existing approaches for indefinite kernels, reviewing the theoretical arguments in favor of stabilization, carefully relating the stabilization problem to the more standard minimization problem, illustrating the different approaches to indefinite kernels with simulations, and providing empirical experiments to evaluate the proposed algorithm.

1.1 Why indefinite kernels with SVMs?

One could consider only the theoretical aspects of kernels and dismiss the need for any further research into indefinite kernels. There exists a lot of efficient positive definite kernels and there is no theoretical proof that, for instance, the `tanh` kernel would be better than the Gaussian RBF kernel. However many applications have similarities that cannot be expressed as a positive definite kernel [3]. We illustrate this with two recent examples. The first is a human evaluation of music similarity, like for the dataset MIREX07 [4] used in this paper. To build the similarity matrix, each human evaluator is asked to judge how similar 2 songs are, using a scale from 0 to 10. The average evaluation for each pair of songs is reported in the similarity matrix. In this example, there is no reason to obtain a positive definite matrix. The second example deals with graph data, that can be extracted from 3d shapes. Computing a graph kernel is a complex task and it is not easy to guarantee that the result will be positive definite. In [5], the author propose a positive definite matching kernel that is presented as an approximation of a previous indefinite version. While the positive version seems to be efficient, who can claim that it is more efficient than the original indefinite one without having a convincing solver able to deal with it to actually compare those kernels? We propose exactly such a solver by expressing the stabilization problem of SVM with indefinite kernels as a standard optimization problem.

1.2 Existing approaches

When dealing with indefinite kernel matrices, there are two main possibilities: either the kernel matrix is

- *G.Loosli is with Clermont Université, Université Blaise Pascal, CNRS, UMR 6158, LIMOS, Aubière, France*
E-mail: gaelle@loosli.fr
- *S.Canu is with Laboratoire LITIS - EA 4108 INSA de Rouen Saint-Étienne-du-Rouvray, France*
E-mail: scanu@insa-rouen.fr
- *C.S.Ong is with NICTA, Canberra, Australia.*
E-mail: cheng-soon.ong@nicta.com.au

changed (via spectrum modification) so that a regular solver can be used, or the solver deals with indefinite matrices directly.

1.2.1 Spectrum modification

We describe here several modifications that can be applied to an indefinite kernel matrix K in order convert it to a positive semidefinite matrix [6], [7]:

- **clip**: the negative part is simply removed (negative eigenvalues cut to 0)
- **shift**: the complete spectrum is shifted until the least eigenvalue is 0
- **flip**: the absolute value of the spectrum is used (the negative part becomes positive)
- **square**: eigenvalues are squared, which is equivalent to using $KK^\top$ instead of K.

Once the matrix is positive definite, a convex optimization algorithm can be used to solve the resulting SVM. However the solution is not based on the true kernel which is a problem for testing new examples.

1.2.2 Solvers accepting indefinite matrices

Several methods are able to deal directly with indefinite matrices. Here we distinguish two kind of solvers: those based on feature-space embedding and those that directly solve the indefinite problem. Feature-space embedding means that each kernel row is considered as an example in the feature space. Hence the indefiniteness is not an issue anymore. Here is a list of feature-space embedding methods that can be found in literature to deal with indefinite kernel matrices:

- **LP-SVM**: proposed in [8], the idea is to use linear programming to solve the SVM problem, since it does not require positive definiteness. [9] proposed a generalised SVM which als includes a linear programming version.
- **P-SVM**: proposed in [10], originally to deal with dyadic data. This solver is based on the usage of $KK^\top$, which is a direct way to consider the kernel rows as features.
- **RVM**: proposed in [11], based on Bayesian inference, it also uses $KK^\top$.

If not using feature-space embedding, the solver has to actually deal with the indefiniteness:

- **non-convex**: the objective function of the problem being non-convex due to the indefinite kernel matrix, it can seem natural to try to minimize it using non-convex optimization. This is done in libSVM solver [12] but is also can be done using difference of convex (DC) techniques [13].
- **spectrum**: there are also methods specifically designed to solve SVM with indefinite kernel, such as IndefiniteSVM [14]–[16]. Here the idea is to do some spectrum modification (clip) while solving. A related approach is IKFD [17], which performs Fisher Discriminant Analysis with the absolute value of the spectrum (flip). Since both these approaches use spectrum modification heuristics, the problem is to treat the test examples in a consistent and theoretically founded fashion.
- **approximate**: instead of choosing a particular way to convert the indefinite kernel, some authors have proposed simultaneously optimising the classifier while finding the best positive semidefinite kernel [15], [18], [19]. These approaches consider the negative parts of the spectrum as noise and aim to correct for it. A further discussion about computational complexity is in section 4.
- **stabilization**: as established in [1], [2] the SVM problem with indefinite kernels are defined in Kreĭn spaces and the solution of the SVM in Kreĭn spaces is the one which stabilizes the cost function. The main problem is then to define what this solution is. This is the point we address in this paper and this solution is provided by the KSVM solver (Kreĭn Support Vector Machine).

There are other related approaches for regression [20] and multiple kernel learning [15], [21] with indefinite kernels.

1.3 Contributions

We consider the problem of training an SVM with an indefinite kernel. The present paper is built on [1], in which the stabilization idea is proposed. However stabilization is a non-standard way to express an optimization problem. With KSVM we bring a well founded and practical solution to the stabilization setting applied to SVM. We show the equivalence between the stabilization problem and a standard convex optimization problem. It turns out that solving the stabilization system (detailed in section 2) can be achieved using a popular heuristic based on the kernel's spectrum modification. On top of providing a solid foundation to spectrum modification, solving the stabilization problem also provides a straight forward way to obtain the solution in the original Kreĭn space. This allows to classify any new point without having to transform it.

We use several illustrative examples to tease apart the difference between the various approaches for indefinite kernels, and include an open source implementation of KSVM. We also demonstrate, in our fully reproducible experiments on empirical data, that KSVM efficiently achieves good results. Overall, not only do we propose in this paper a full study of the KSVM, but we also intend to convince the reader that understanding the impact of working in a Kreĭn space is the key to solving learning problems with indefinite kernels.

2 SVM IN KREĬN SPACES

In this section we review the background on Kreĭn spaces and establish the stabilization system that has to be solved to train an SVM in a Kreĭn space. We expand upon the theoretical properties of Reproducing Kernel Kreĭn Spaces as first presented to the machine learning community in [1].

2.1 SVM as a projection

The success of SVMs in numerous application areas has been made possible thanks to the existence of stable, efficient and accurate numerical algorithms. This is due to the fact that the problem of margin maximization in a Reproducing Kernel Hilbert Space (RKHS) can be cast as a quadratic program. We introduce here a slightly non-standard formulation of the SVM which is equivalent to the standard C-SVM. Let $x_i \in \mathcal{X}, i \in \{1, \dots, \ell\}$ be ℓ training points in the input space $\mathcal{X}$, along with their label $y_i \in \{-1, 1\}$ representing the class each point belongs to in a classification problem. The input space $\mathcal{X}$ is often considered to be $I\!R^d$, but can be any space due to the kernel trick. For a given positive C, SVM is the minimum of the following regularized empirical risk functional

$$J_C(f, b) = \tfrac{1}{2}\|f\|^2_{\mathcal{H}} + C \sum_{i=1}^{\ell} \max\big(0, 1 - y_i(f(x_i) + b)\big). \quad (1)$$

Using its solution $(f^\star_C, b^\star_C) := \arg\min J_C(f, b)$ we can introduce $\tau = \sum_{i=1}^{\ell} \max\big(0, 1 - y_i(f^\star_C(x_i) + b^\star_C)\big)$ and the associated convex quadratic program (QP)

$$\begin{cases} \displaystyle\min_{f \in \mathcal{H}, b \in I\!R} & \tfrac{1}{2}\|f\|^2_{\mathcal{H}} \\ \text{s.t.} & \displaystyle\sum_{i=1}^{\ell} \max\big(0, 1 - y_i(f(x_i) + b)\big) \leq \tau \ . \end{cases} \quad (2)$$

This QP admits as unique solution $(f^\star_\tau, b^\star_\tau, \lambda^\star_\tau)$ where $\lambda^\star_\tau$ denotes the value of the Lagrange multiplier of the constraint at the optimum. It turns out that these two problems (1) and (2) are equivalent in the sense that, for well chosen C and τ, they provide the same unique solution $(f^\star_C, b^\star_C) = (f^\star_\tau, b^\star_\tau)$. Indeed, J_C is convex as the sum of two convex functionals and by setting $C = \lambda^\star_\tau$, its regularization parameter can be interpreted as the optimal Lagrange multiplier of (2), so that the solution of (1) verifies the KKT conditions of (2) and the solution of (2) verifies, thanks to its stationary condition, the optimality condition for being a solution of (1).

The QP (2) can be also seen as the problem of retrieving the orthogonal projection of the null function in $\mathcal{H}$ onto the convex feasible set. This formulation allows us to define the so-called "variational inequality" characterization of projection (equivalent in this case to the optimality condition for f being the solution of problem (2)). The bias b play a particular role since it is not explicitly involved in the projection. A way to treat it, and thus to preserve the uniqueness of the solution, is to include an additional equation in the definition of the convex set we are projected on. This equation is the optimality condition regarding b that is $0 \in \partial_b H(f, b)$ where ∂_b denote the sub differential with respect to b and

$$H(f, b) = \sum_{i=1}^{\ell} \max\big(0, 1 - y_i(f(x_i) + b)\big)$$

Definition 2.1 (SVM as a projection): Let $\mathcal{H}$ be a RKHS. For a given set $\mathcal{S}$,

$$\mathcal{S} = \big\{f \in \mathcal{H}, b \in I\!R \mid H(f, b) \leq \tau \text{ and } 0 \in \partial_b H(f, b)\big\},$$

the SVM is the unique $(f, b) \in \mathcal{S}$ such that

$$\forall (g, a) \in \mathcal{S}, \quad \langle f, f - g\rangle_{\mathcal{H}} \ \leq \ 0.$$

Since it does not involve any norm, this way of defining SVM using a projection regularization principle[1] can be used as it is when dealing with indefinite kernels in Kreĭn spaces. However, as we will see in the following section, this will lead to a problem of finding a stationary point instead of a minimum. We call the problem of finding a stationary point "stabilization".

2.2 A quadratic program to solve SVM in Kreĭn spaces using stabilization

In this section, Reproducing Kernel Kreĭn Spaces (RKKS) are briefly introduced to allow the definition of SVMs in this framework as a projection. Then this definition is recast as a stabilization problem. It was shown in [1] that from the function space point of view, positive semi-definiteness is *not* a requirement, and in fact the representer theorem is also valid for RKKS.

2.2.1 Reproducing Kernel Kreĭn Space

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

Definition 2.2 (Inner Product, [23]): 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 2.3 (Kreĭn space, [24]): 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 $\mathcal{K}$, with $f_+ \in \mathcal{H}_+$ and $f_- \in \mathcal{H}_-$, such that

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

If $\mathcal{H}_+$ and $\mathcal{H}_-$ are RKHS, $\mathcal{K}$ is a reproducing kernel Kreĭn spaces (RKKS). In this case the uniqueness

[1] Note that projection is a well known regularization mechanism (see for instance [22] and related references), allowing to choose a unique and stable solution from the feasible set $\mathcal{S}$.

of the functional decomposition (the nature of the RKHSs $\mathcal{H}_+$ and $\mathcal{H}_-$) is not guaranteed. In [1], Proposition 6, the reproducing property is shown: in a RKKS $\mathcal{K}$, there is a unique symmetric kernel $k(x,x')$ with $k(x,.) \in \mathcal{K}$ such that the reproducing property holds (for all $f \in \mathcal{K}, f(x) = \langle f, k(x,.)\rangle_{\mathcal{K}}$) and $k = k_+ - k_-$ where k_+ and k_- are the reproducing kernels of the RKHSs $\mathcal{H}_+$ and $\mathcal{H}_-$. Furthermore, for any symmetric nonpositive kernel k that can be decomposed as the difference of two positive kernels k_+ and k_-, a RKKS can be associated to it.

2.2.2 SVM in RKKS

The definition of a proper SVM in RKKS requires an adaptation from the classical SVM in RKHS given by (2) since the norm $\|f\|$ is not defined in Kreĭn spaces. As previously remarked, the minimization of a norm can be seen as a projection. This interpretation in terms of projection still holds in Kreĭn spaces and can be used as a regularization mechanism [25], [26]. This allows to define SVM in RKKS (as it can be in Hilbert spaces) as the orthogonal projection of the null element onto

$$\mathcal{S} = \Big\{ f \in \mathcal{K}, b \in I\!R \;\Big|\; H(f,b) \le \tau \;\text{ and }\; 0 \in \partial_b H(f,b) \Big\}$$

Definition 2.4 (SVM in a RKKS): Let $\mathcal{K}$ be a RKKS. For a given set $\mathcal{S}$, the SVM is the unique $(f,b) \in \mathcal{S}$ such that

$$\forall g \in \mathcal{S}, \quad \langle f, f-g\rangle_{\mathcal{K}} \;\le\; 0.$$

As claimed in [25], 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 *stabilize* 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

$$\begin{cases} \operatorname*{stab}_{f\in\mathcal{K}, b\in I\!R} & \frac{1}{2}\langle f, f\rangle_{\mathcal{K}} \\ \text{s.t.} & \displaystyle\sum_{i=1}^{\ell} \max\big(0, 1 - y_i(f(x_i)+b)\big) \le \tau\,. \end{cases} \tag{3}$$

where stab means *stabilize*.

The literature on convex optimization [27], [28] 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) have the same first order conditions of optimality.

2.2.3 An illustrative example

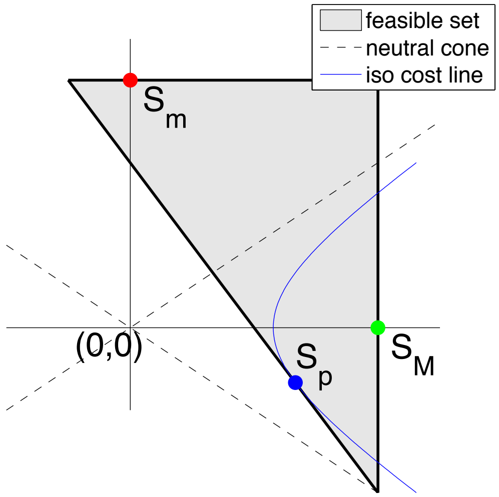

Fig. 1. The three solutions of the illustrative example.

To illustrate this idea of projection in a Kreĭn space we propose to consider the following cost function $J(w_1,w_2) = w_1^2 - w_2^2$ (that can be seen as an inner product in a Kreĭn space) together with the feasible set

$$\mathcal{S} = \left\{ (w_1,w_2) \;\middle|\; \begin{array}{rl} w_1 & \le 2 \\ w_2 & \le 3 \\ w_1 + \frac{1}{2}w_2 & \ge 1 \end{array} \right\}.$$

Regardless on whether we want to maximize, minimize or stabilize the cost function $J(w_1,w_2)$, the Lagrangian of the problem is the same, namely:

$$\begin{aligned} \mathcal{L}(w_1,w_2,\lambda_1,\lambda_2,\lambda_3) = w_1^2 - w_2^2 + \\ \lambda_1(w_1-2) + \lambda_2(w_2-3) - \lambda_3(w_1 + \tfrac{1}{2}w_2 - 1). \end{aligned}$$

Thus, a stationary point of this problem satisfies

$$\begin{gathered} \nabla_{\mathbf{w}}\mathcal{L}(w_1,w_2,\lambda_1,\lambda_2,\lambda_3) = 0 \\ \Leftrightarrow \\ \begin{cases} 2w_1 + \lambda_1 - \lambda_3 = 0 \\ -2w_2 + \lambda_2 - \frac{1}{2}\lambda_3 = 0. \end{cases} \end{gathered}$$

The associated KKT conditions admit three solutions represented figure 1. The first solution minimizing $J(w_1,w_2)$ is $S_m = (0,3)$, the second solution maximizing $J(w_1,w_2)$ is $S_M = (2,0)$ while the third solution $S_p = (\frac{4}{3}, \frac{-2}{3})$ is the Kreĭn space projection of $(0,0)$ onto the feasible set. Indeed, $\forall \mathbf{g} = (g_1,g_2) \in \mathcal{S}$, $\langle S_p, S_p - \mathbf{g}\rangle_{\mathcal{K}} = \frac{4}{3}(\frac{4}{3} - g_1) - \frac{-2}{3}(\frac{-2}{3} - g_2) \le 0$ illustrating the fact that the projection of $(0,0)$ onto the feasible set is a stationary point of the Lagrangian.

2.3 The dual quadratic program using stabilization

2.3.1 A Saddle Point

In the following we need to transform the stationary point search into a min-max search. This is easy to write if the stationary point is a saddle point, a fact we prove below.

Proposition 2.1: The stationary point is a saddle point.

Proof: To show our point, we follow [25, section 6.3.1]. Let consider the quadratic cost function

$$I(a,b) = \left[\begin{array}{cc} a^\top & b^\top \end{array} \right] \left[\begin{array}{cc} A & B \\ B^\top & C \end{array} \right] \left[\begin{array}{c} a \\ b \end{array} \right] \quad (4)$$

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 problem is a stationary point. Let's say that one wants to minimize $I(a,b)$ through the choice of a and maximize it through the choice of b, then there are two different strategies that can be applied: either the max-min problem ($\max_b \min_a I(a,b)$) or the min-max problem ($\min_a \max_b I(a,b)$). As stated in [25, 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 positive definite and* C *is negative definite.*

Now we show that our cost function $I1(f_+, f_-)$ can be written such that it is possible to identify A and C and deduce that the stationary point is a saddle point. Let

$$I1(f_+, f_-) = \tfrac{1}{2}\langle f_+, f_+\rangle_{\mathcal{H}^+} - \tfrac{1}{2}\langle f_-, f_-\rangle_{\mathcal{H}^-}. \quad (5)$$

From the Kreĭn space decomposition we have

$$f(.) = f_+(.) - f_-(.)$$

with

$$f_+(.) = \sum_i \beta_i y_i k_+(x_i, .)$$

and

$$f_-(.) = \sum_i \beta_i y_i k_-(x_i, .).$$

Hence eq. (5) can be expressed as

$$I1(\beta) = \frac{1}{2}\beta^\top G_+ \beta - \frac{1}{2}\beta^\top G_- \beta,$$

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 $G = G_+ - G_-$. We 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 $a = U^\top \beta = [b_+; b_-] = [U_-^\top \beta; U_+^\top \beta]$.

$$\begin{aligned} I1(\beta) &= \frac{1}{2}\beta^\top U_+ D_+ U_+^\top \beta - \frac{1}{2}\beta^\top U_- D_- U_-^\top \beta \\ I1(b_+, b_-) &= \frac{1}{2} b_+^\top D_+ b_+ - \frac{1}{2} b_-^\top D_- b_- \\ I1(b_+, b_-) &= \left[\begin{array}{cc} b_+^\top & b_-^\top \end{array} \right] \left[\begin{array}{cc} D_+ & 0 \\ 0 & D_- \end{array} \right] \left[\begin{array}{c} b_+ \\ b_- \end{array} \right] \end{aligned} \quad (6)$$

From this we can identify with eq.(4) and see by definition that $A = D_+$ is positive definite and $C = D_-$ is negative definite. This shows that the stationary point of our problem is a saddle point. □

Since the stationary point we are looking for is a saddle point, we can write it as a min-max or as a max-min problem, which we use in the following.

2.3.2 An equivalent loss function

To characterize the solutions of indefinite SVM in RKKS given by (3), it would be useful to define the stabilization problem as a unique loss function $J(f)$, that would be composed of $J_1(f)$, the term to be stabilized and $J_2(f)$, the good classification constraints. However, going from (3) to $J(f)$ is not a standard manipulation in the case of stabilization, in particular when regarding the sign of $J_2(f)$. In the following, we decompose the stabilization problem into a min-max problem (see section 2.3.1). $J(f)$ becomes then $J(f_+, f_-)$. The problem is then minimized according to f_+ and maximized according to f_-.

$$\left\{ \begin{array}{ll} \min\limits_{f_+,b,\xi} \max\limits_{f_-,b,\xi} & \frac{1}{2}\langle f_+, f_+\rangle_{\mathcal{H}_+} - \frac{1}{2}\langle f_-, f_-\rangle_{\mathcal{H}_-} + C\sum\limits_{i=1}^{\ell} \xi_i \\ \text{s.t.} & y_i(f_+(x_i) - f_-(x_i) + b) \geq 1 - \xi_i \\ \text{and} & \xi_i \geq 0 \quad \forall i \in [1..\ell] \end{array} \right. \quad (7)$$

Step 1: Minimization according to f_+: First we fix f_- in eq. (7) and write the equivalent loss function for f_+. In that case, we only have a standard minimization problem.

$$\left\{ \begin{array}{lll} \min\limits_{f_+,b,\xi} & \frac{1}{2}\langle f_+, f_+\rangle_{\mathcal{H}_+} + C\sum\limits_{i=1}^{\ell} \xi_i & \\ \text{s.t.} & y_i(f_+(x_i) - f_-(x_i) + b) \geq 1 - \xi_i & \forall i \in [1..\ell] \\ \text{and} & \xi_i \geq 0 & \forall i \in [1..\ell] \end{array} \right. \quad (8)$$

Using the KKT conditions, we can write the associated loss function as:

$$\begin{aligned} J_p(f_+, b) = \;& \tfrac{1}{2}\langle f_+, f_+\rangle_{\mathcal{H}_+} + \sum_{i=1}^{\ell} \beta_i \\ & - \sum_{i=1}^{\ell} \beta_i (y_i(f_+(x_i) - f_-(x_i) + b) \\ \text{with} \;& 0 \leq \beta_i \leq C \end{aligned} \quad (9)$$

Step 2: Maximization according to f_-: First we fix f_+ in eq. (7) and write the equivalent loss function for f_-. In that case, we only have a standard maximization problem.

$$\left\{ \begin{array}{lll} \min\limits_{f_-,b,\xi} & -\frac{1}{2}\langle f_-, f_-\rangle_{\mathcal{H}_-} + C\sum\limits_{i=1}^{\ell} \xi_i & \\ \text{s.t.} & y_i(f_+(x_i) - f_-(x_i) + b) \geq 1 - \xi_i & \forall i \in [1..\ell] \\ \text{and} & \xi_i \geq 0 & \forall i \in [1..\ell] \end{array} \right. \quad (10)$$

Using the KKT conditions, we can write the associated loss function as:

$$\begin{aligned} J_n(f_-,b) = & -\tfrac{1}{2}\langle f_-,f_-\rangle_{\mathcal{H}_-} - \sum_{i=1}^{\ell}\gamma_i \\ & + \sum_{i=1}^{\ell}\gamma_i(y_i(f_+(x_i) - f_-(x_i) + b) \\ \text{with} \quad & 0 \le \gamma_i \le C \end{aligned} \tag{11}$$

Step 3: Loss function associated to the min-max problem: We now define J_d from (9) and (11) as follows:

$$\begin{aligned} J_d(f_+,f_-,b) = & \ J_p(f_+,b) + J_n(f_-,b) \\ = & \ \frac{1}{2}\langle f_+,f_+\rangle_{\mathcal{H}_+} - \frac{1}{2}\langle f_-,f_-\rangle_{\mathcal{H}_-} \\ + & \sum_{i=1}^{\ell}\beta_i - \sum_{i=1}^{\ell}\gamma_i \\ - & \sum_{i=1}^{\ell}\beta_i(y_i(f_+(x_i) - f_-(x_i) + b) \\ + & \sum_{i=1}^{\ell}\gamma_i(y_i(f_+(x_i) - f_-(x_i) + b) \end{aligned} \tag{12}$$

and claim that (12) is the loss function associated to (7). We introduce $\alpha_i = \beta_i - \gamma_i$ and we observe the possible values of α_i in table (1):

	$\beta_i = 0$	$0 < \beta_i < C$	$\beta_i = C$
$\gamma_i = 0$	0	$0 < \alpha_i < C$	$\alpha_i = C$
$0 < \gamma_i < C$	$-C < \alpha_i < 0$	$-C < \alpha_i < C$	$0 < \alpha_i < C$
$\gamma_i = C$	$\alpha_i = -C$	$-C < \alpha_i < 0$	0

TABLE 1
Possible values of α_i

The loss function (12) is then written as:

$$\begin{aligned} J_d(f_+,f_-,b) = & \ \frac{1}{2}\langle f_+,f_+\rangle_{\mathcal{H}_+} - \frac{1}{2}\langle f_-,f_-\rangle_{\mathcal{H}_-} \\ + & \sum_{i=1}^{\ell}\alpha_i \\ - & \sum_{i=1}^{\ell}\alpha_i(y_i(f_+(x_i) - f_-(x_i) + b) \end{aligned} \tag{13}$$

Step 4: Back to $f \in \mathcal{K}$: From eq. (13) it is easy to write the loss function associated to the stabilization problem for which we want to write the representer theorem:

$$\begin{aligned} J(f,b) = & \ \frac{1}{2}\langle f,f\rangle_{\mathcal{K}} + \sum_{i=1}^{\ell}\alpha_i \\ - & \sum_{i=1}^{\ell}\alpha_i(y_i(f(x_i) + b)) \end{aligned} \tag{14}$$

2.3.3 The representer theorem for SVM in RKKS

The stationnary point of $J(f,b)$ is given as usual by annihilating its gradients.

$$\begin{cases} \nabla_f J(f,b) = f(.) - \sum_{i=1}^{\ell}\alpha_i y_i k(x_i,.) = 0 \\ \nabla_b J(f,b) = -\sum_{i=1}^{\ell}\alpha_i y_i = 0 \end{cases} \tag{15}$$

This results in the following:

$$\begin{cases} \qquad f(.) = \sum_{i=1}^{\ell}\alpha_i y_i k(x_i,.) \\ \text{with} \qquad \sum_{i=1}^{\ell}\alpha_i y_i = 0 \\ \text{and} \qquad -C \le \alpha_i \le C \qquad \forall i \in [1..\ell] \end{cases} \tag{16}$$

2.3.4 Dual stabilization problem

To obtain the dual stabilization problem, we substitute eq.(16) in the loss function J:

$$J(f,b) = -\frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\alpha_i\alpha_j y_i y_j k(x_i,x_j) + \sum_{i=1}^{\ell}\alpha_i \tag{17}$$

and we deduce the dual stabilization quadratic problem:

$$\begin{cases} \underset{\alpha}{\text{stab}} & -\frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\alpha_i\alpha_j y_i y_j k(x_i,x_j) + \sum_{i=1}^{\ell}\alpha_i \\ \text{with} & \sum_{i=1}^{\ell}\alpha_i y_i = 0 \\ \text{and} & -C \le \alpha_i \le C \quad \forall i \in [1..\ell] \end{cases} \tag{18}$$

In the following section, we derive an other path to go from the primal stabilization problem (3) to the dual one (18), using much more standard tools. The reason why we propose two approaches is two fold: on the one hand it permits to confirm our proposition for the dual of a stabilization problem. On the other hand, the second path given in section 3 provides the equations used in the algorithm KSVM, solving the SVM in Kreĭn spaces.

3 Alternative path from Primal to Dual Stabilization Problem

The algorithm (1) presented in section 4, called KSVM, is based on the equivalence that is established in this section. We show that the stabilization problem (3) can be written as a minimization problem using a semi-definite kernel matrix. We give a series of relations between the stabilization variables and the minimization ones and we also use them to write the dual of our stabilization problem. Overall we consider 4 distinct optimization problems and we establish their equivalence in terms of optimality conditions:

1) Primal stabilization problem (Equation (3)): we proved in section 2.3.1 that the stationary point is a saddle point, which means that the problem can can transformed into a min-max or a max-min problem indifferently. This allows to go to the second step.
2) Primal minimization problem (Equation (19)): this standard problem leads directly to the next one via convex duality.
3) Dual maximization problem (Equation (20)): we define in section 3.3.1 a matrix transition P that is used to propose the final problem.
4) Dual stabilization problem (Equation (24))

Note that in the following, we use β for coefficients in the primal, α for coefficients in the dual RKKS and $\tilde\alpha$ for coefficients in the dual RKHS.

3.1 Equivalence between Stabilization and Minimization

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. We first write equation (3) according to f_+ and f_-, since we have $f_\mathcal{K} = f_+ + f_-$ and $\langle f, f\rangle_\mathcal{K} = \langle f_+, f_+\rangle_{\mathcal{H}^+} - \langle f_-, f_-\rangle_{\mathcal{H}^-}$.

$$\begin{cases} \displaystyle\min_{f_+\in\mathcal{H}^+, b\in I\!R} \max_{f_-\in\mathcal{H}^-} & \frac{1}{2}\langle f_+, f_+\rangle_{\mathcal{H}^+} - \frac{1}{2}\langle f_-, f_-\rangle_{\mathcal{H}^-} \\ \text{s.t.} & \displaystyle\sum_{i=1}^{\ell} \max\big(0, 1 - y_i(f_+(x_i) + f_-(x_i) + b)\big) \le \tau \end{cases}$$

From here, is it possible to change the maximization part into a minimization as follows:

$$\begin{cases} \displaystyle\min_{f_+\in\mathcal{H}^+, f_-\in\mathcal{H}^-, b\in I\!R} & \frac{1}{2}\langle f_+, f_+\rangle_{\mathcal{H}^+} + \frac{1}{2}\langle f_-, f_-\rangle_{\mathcal{H}^-} \\ \text{s.t.} & \displaystyle\sum_{i=1}^{\ell} \max\big(0, 1 - y_i(f_+(x_i) + f_-(x_i) + b)\big) \le \tau \end{cases}$$

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

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

Using this notation, we obtain

$$\begin{cases} \displaystyle\min_{\tilde f\in\tilde{\mathcal{K}}, b\in I\!R} & \frac{1}{2}\langle \tilde f, \tilde f\rangle_{\tilde{\mathcal{K}}} \\ \text{s.t.} & \displaystyle\sum_{i=1}^{\ell} \max\big(0, 1 - y_i(\tilde f(x_i) + b)\big) \le \tau \end{cases} \tag{19}$$

which is the SVM formulation given in equation (2).

3.2 Dual Optimization Problem

By using standard methods of Lagrange duality, the dual optimization problem corresponding to Equation (19) is given by

$$\begin{cases} \displaystyle\max_{\tilde\alpha} & -\frac{1}{2}\tilde\alpha^\top \tilde G \tilde\alpha + \tilde\alpha^\top \mathbf{1} \\ \text{subject to} & \tilde\alpha^\top \mathbf{y} = 0 \\ \text{and} & 0 \le \tilde\alpha_i \le C \qquad \forall i \in [1..\ell] \end{cases} \tag{20}$$

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

3.3 An Equivalent stabilization Problem in its Dual Form

To write the dual stabilization problem, we define projection operators and transition matrices that are used to relate α in the dual RKKS and $\tilde\alpha$ in the dual RKHS.

3.3.1 Transition matrix between α and $\tilde\alpha$

Definition 3.1 (Fundamental decomposition of $\mathcal{K}$): [25, Definition 2.2.1, remarks]. We define two projection operators $\mathcal{P}_+$ and $\mathcal{P}_-$ such that

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

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

$$x = x_+ + x_-, \quad \text{where}$$

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

Proposition 3.1: Matrices P_+ and P_- are given by the eigen-decomposition of the matrix G.

Proof: Using the same eigen-decomposition as for eq. (6), $G = UDU^\top$, we can write

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

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

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

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

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 transition matrix such that $P = P_+ - P_- = USU^\top$ with

$$S = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$$

Let's decompose α according to P_+ and P_- and deduce the decomposition of $\tilde\alpha$:

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

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

3.3.2 From dual maximization to dual stabilization

We now use the relation between α and $\tilde{\alpha}$ in problem 3, eq (20).

$$\left\{\begin{array}{ll} \max\limits_{\alpha} & -\frac{1}{2}\alpha^\top P\tilde{G}P\alpha + \alpha^\top P\mathbf{1} - b\alpha^\top P\mathbf{y} \\ \text{with} & 0 \leq (P\alpha)_i \leq C \quad \forall i \in [1..\ell] \end{array}\right. \tag{21}$$

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 $P_-GP_- = G_-$:

$$\left\{\begin{array}{ll} \max\limits_{\alpha_+,\alpha_-} & -\frac{1}{2}\alpha_+^\top G_+\alpha_+ + \frac{1}{2}\alpha_-^\top G_-\alpha_- + \alpha_+^\top\mathbf{1} - \alpha_-^\top\mathbf{1} \\ & -\mu(\alpha_+ - \alpha_-)^\top\mathbf{y} \\ \text{with} & 0 \leq (\alpha_+ - \alpha_-)_i \leq C \quad \forall i \in [1..\ell] \end{array}\right. \tag{22}$$

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

$$\left\{\begin{array}{ll} \max\limits_{\alpha_+}\min\limits_{\alpha_-} & -\frac{1}{2}\alpha_+^\top G_+\alpha_+ - \frac{1}{2}\alpha_-^\top G_-\alpha_- + \alpha_+^\top\mathbf{1} + \alpha_-^\top\mathbf{1} \\ & -\mu(\alpha_+ + \alpha_-)^\top\mathbf{y} \\ \text{with} & 0 \leq \alpha_{+i} \leq C_+ \quad \forall i \in [1..\ell] \\ \text{and} & -C_- \leq \alpha_{-i} \leq 0 \quad \forall i \in [1..\ell] \\ \text{and} & C_+ + C_- = C \end{array}\right. \tag{23}$$

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

$$\left\{\begin{array}{lll} \operatorname*{stab}\limits_{\alpha} & -\frac{1}{2}\alpha^\top G\alpha + \alpha\mathbf{1} - \mu\alpha^\top\mathbf{y} & \\ \text{and} & -C_- \leq \alpha_i \leq C_+ & \forall i \in [1..\ell] \end{array}\right. \tag{24}$$

System (24) is similar to the dual (18), up to constants.

4 Practical algorithm: KSVM and KSVM-L

The resulting algorithm, that computes the solution of the stabilization problem by solving the equivalent SVM dual minimization problem is given by algorithm (1) and named KSVM (for Kreĭn SVM). We denote G the kernel matrix such that $G(i,j) = y_iy_jk(x_i,x_j)$, and describe our proposed algorithm in Algorithm (1).

Algorithm 1 SVM solver for indefinite kernels in Kreĭn spaces (KSVM)

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

This solver produces an exact solution for the stabilization problem. Its main weakness is that it requires the user to pre-compute the whole kernel matrix and to decompose it into eigenvectors/eigenvalues. The other point to mention 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 [29]–[32]. This low rank adaptation is referred as KSVM-L.

Note that KSVM is in practice quite similar to the indefinite SVM proposed in [15], [18], even though the reasoning is very different: in [15], [18], the idea is to learn a semi-definite positive kernel matrix from the initial indefinite matrix during training. This leads to a convex conic optimization problem, which has 2 major drawbacks: it is large and it produces solutions that over-fit. This is the reason why the authors propose to restrict the possible surrogate matrices to be a spectrum modification of the original matrix. This spectrum modification is learned using a second-order cone program [18] and a bundle method [15] so the solution would not necessarily be the same as KSVM's. During test time, [18] solves a QCQP to transform the test samples with the same spectrum modification as the training samples. Concerning the test part, it differs from KSVM only in the fact that they use the spectrum modification information to transform the test samples while we use the same information to get the solution back into the original Kreĭn space. In addition, since KSVM can leverage on existing efficient SVM solvers, it is computationally much faster than indefinite SVM that needs to solve a second order cone program.

5 Experiments

This part contains a series of experiments that show that our approach leads to better results than the previous approaches. In the first subsection, we propose a simple visualization of the solutions given by stabilization and minimization on a 2d problem with a linear indefinite kernel. The solutions provided by various algorithms on the checkerboard data are also presented. The next subsection presents an extensive experimental study on various datasets in order to compare the performance of each of the previous approaches to deal with indefinite kernels (methods that require to modify test data are excluded). Finally we compare the experimental complexity of the different solvers.

5.1 Solvers

We perform an experimental comparison of the following methods [2]:

- KSVM/KSVM-L ([2] and this paper), P-SVM [10], [34], Cut, Shift and Flip (usual heuristics).
- IndefiniteSVM is the Matlab implementation provided by the authors [15]
- libSVM [12] provides a Matlab connection
- RVM is the Matlab implementation provided by the authors [35]
- LP-SVM is based on CPLEX [36]

5.2 Illustrations

The intuition behind the stabilization problem is not straight forward. It requires us to think about the meaning of the negative part of the space. A very interesting viewpoint is introduced in [3], 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.

Stabilization vs Minimization in 2D

The goal of this experiment is to provide a visual hint on the difference between minimizing and stabilizing indefinite SVM. The space is 2d, and the kernel is linear yet indefinite:

$$k(x,y) = x_1y_1 - x_2y_2 \tag{25}$$

Figure 2 shows the minimization decision function (in pink) and the stabilization decision function (in green). We observe that both have a zero training error. However the stabilization decision function has a larger margin.

To go a little further in this example, the same setting was run two hundred times, with randomly generated training set, of random size (between 10 and 100 for each class). For each couple of classifier (minimizer and stabilizer), margins (in the Krein space - M_K - and in the Euclidean space - M_E) and cost function value (C_O) at the optimum are computed. On top of that, we count the number of times that the solutions are very close (same margin value, same optimal cost value). We report in table 2 the average results presented as the gain of stabilization over minimization:

$$M_K = 100 \times \text{mean}((M_K^{Stab} - M_K^{Min})/M_K^{Min})$$

$$M_E = 100 \times \text{mean}((M_K^{Stab} - M_K^{Min})/M_K^{Min})$$

$$C_O = 100 \times \text{mean}((C_O^{Stab} - C_O^{Min})/C_O^{Min})$$

M_K	M_E	C_O (variance)
+42.40 %	+131.11 %	+32.13 (± 42.09) %

TABLE 2
Average gain of stabilization over minimization in terms of margins, along with the average difference of cost function optimal value. In 14.5% of the tested datasets, the margins are similar for both solvers.

The results illustrate that margins are significantly larger, in both Euclidean and Kreĭn spaces, when stabilizing, as shown in figure 2. The margins resulting from stabilization are always larger than when performing minimization, even though on average the stabilization solution has a higher cost function value. This shows that lower cost functions do not correlate with larger margins, indicating that minimization is not the right objective. Furthermore, in 17% of the tests the stabilization solution is lower than the minimization solution, demonstrating that the minimization procedure is stuck in a local minimum that is not globally optimal.

Margin maximization and Kreĭn space

In Kreĭn spaces, the numerical margin remains well defined. However, the notion of geometric margin is not yet clear. Indeed, in the usual Euclidean framework, the geometric margin is defined from the associated norm, no longer available in non-Euclidean geometry (indefinite space).

More precisely, in the positive linear separable case, the margin maximization problem can be written as follows

$$\begin{cases} \max\limits_{v,a} & m \\ \text{s.t.} & \min\limits_{i\in[1,\ell]} \dfrac{|\langle v,x\rangle + a|}{\langle v,v\rangle^{\frac{1}{2}}} \geq m \end{cases}$$

where (v,a) defines the decision function. Defining $w = \frac{v}{m\sqrt{\langle v,v\rangle}}$ and $b = \frac{a}{m\sqrt{\langle v,v\rangle}}$, we can rewrite this system

$$\begin{cases} \max\limits_{w,b} & \dfrac{1}{\langle w,w\rangle} \\ \text{s.t.} & y_i\big(\langle w,x_i\rangle + b\big) \geq 1 \quad \forall i \end{cases}$$

And then this maximization problem is converted to the equivalent minimization problem on $\langle w,w\rangle$.

Obviously, going from the margin maximization to the scalar product minimization relies several times on the fact that the scalar products are positive. It is

[2] The pure Matlab solvers corresponding to KSVM, KSVM-L, P-SVM and the spectrum modification methods, are implemnted using the SimpleSVM Toolbox [33] and are available at http://gaelle.loosli.fr/research/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2477830, IEEE Transactions on Pattern Analysis and Machine Intelligence

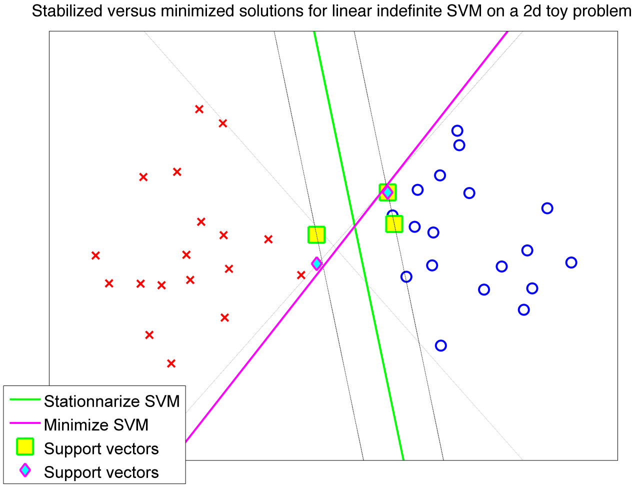

Fig. 2. Illustration of the solutions provided by minimizing or stabilizing the cost function of a 2d indefinite problem with a linear kernel.

easy to see that both the variable change from v to w and the problem conversion from $\frac{1}{\langle w,w\rangle}$ to $\langle w,w\rangle$ face signs issues which prevent from writting max or min problems. This enlights the fact that minimization in the indefinite case will not provide a large margin.

Now concerning the indifinite case, stabilization maximizes the margin in Kreĭn space. The difficulty is that it is in general not possible to reduce a non-Euclidean geometry into Euclidean geometry, and therefore it is not possible to reduce the margin in Kreĭn space into a margin in Hilbert space.

Results on checkerboards

Here we show the behaviors of the learning methods on a checkerboard patterned dataset. Figures 3 and 4 give an illustration of each considered method's result. For the classical heuristics *Cut*, *Shift* and *Flip* (which consists in modifying the kernel spectrum, respectively in removing negative eigenvalues, increasing those until they are all positive and taking the absolute value), we present the decision function with and without including the test set in the kernel modification. The test set is composed of a regular grid (41×41) on all the plotted space, those points being used to actually plot the decision boundaries.

For figure 3, 96 training points are used, while 400 training points are used for figure 4. Visually, it is quite easy to observe that some methods have difficulties to learn the data shape: IndefiniteSVM, P-SVM, libSVM, Cut and Shift (with or without transforming the test set actually). Although RVM fails regularly for the small dataset case it behaves well otherwise. Flip with test set transformation gives results quite similar to KSVM. LP-SVM and KSVM are visually comparable.

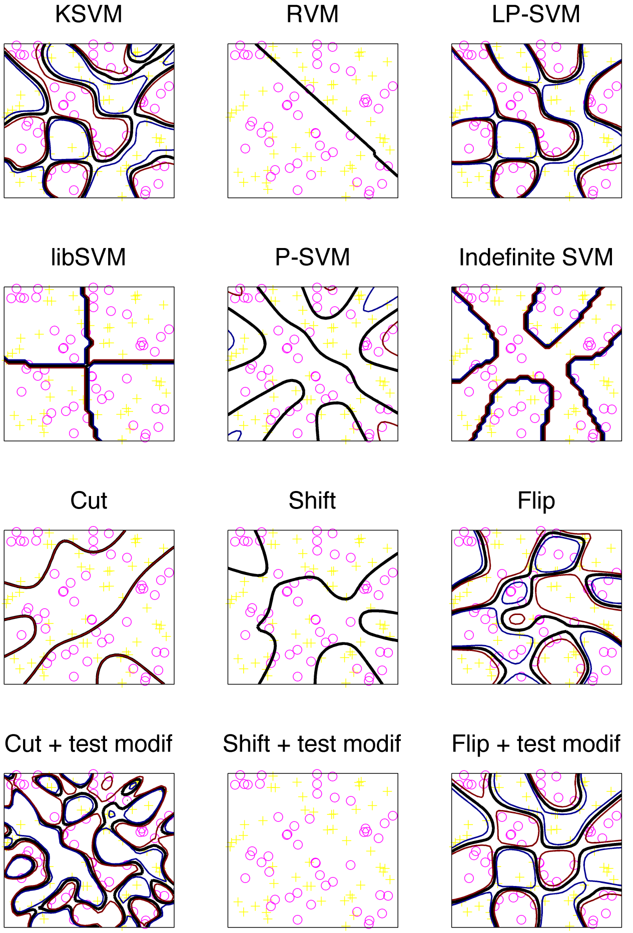

Fig. 3. Training 2D checkers on 96 training points with different methods. The least eigenvalue of the tanh kernel is -13.11.

5.3 Stabilization for indefinite kernels and dissimilarity kernels

This large experiment is meant to give an overview of the ability of KSVM to solve open problems in a large variety of application fields. Indeed, it is quite easy to find datasets leading to indefinite kernel matrices. From biological field (with graph kernels, string kernels...) to vision, passing by human based similarities (for instance when humans are asked to evaluate something they ear), indefiniteness arises naturally in many situations. From other interesting studies on learning with indefinite kernels (for instance [37]–[39]), we have collected 14 datasets, plus 2 UCI classics. Table 3 gives a description of each dataset.

Experimental setting

For each dataset, we have run 20 times the following procedure: a random split to produce a training and a testing set, a 5-fold cross validation to tune each parameter (the number of parameters depending on the method) on the training set, and the evaluation on the testing set.

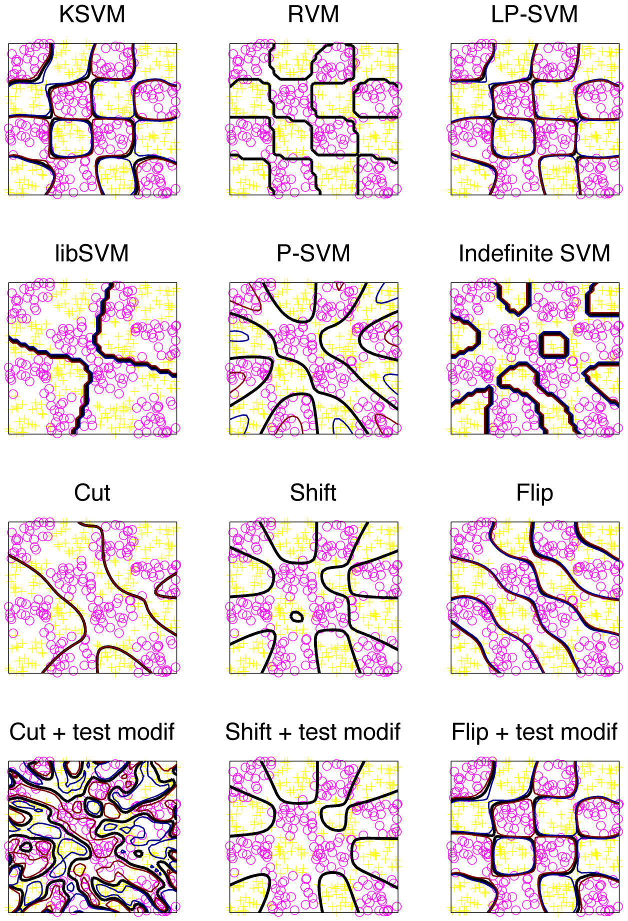

Fig. 4. Training 2D checkers on 400 training points with different methods. The least eigenvalue of the tanh kernel is -59.73.

Results

Table 4 gives average error rates and standard deviation of KSVM-L, RVM, LP-SVM, and for comparison the best published results found in literature among [4], [37]–[40]. Most of the time, the experimental settings are comparable to ours. We observe that KSM-L is always more accurate or close to the best published. The interesting point is this consistency among the different datasets, which is a quality that the other methods cannot claim.

5.4 Computation time

In this section we show the result of a simple experiment dedicated to the training and testing time. Figure 5 shows in log scales the evolution of the training and testing time together depending on the training set size. The test set size is constant to 1000.

The experiment shows that the most efficient method is KSVM-L, which is an efficient implementation of KSVM (using partial eigen-decomposition). libSVM solver is faster (note that we used the compiled code and not native Matlab) but the slope of its curve is higher.

Dataset	p	n	k	Type
Balls3D	0.8	200	2	synthetic dissimilarity
diabetes	0.5	768	2	tanh kernel
a1a	0.8	1605	2	tanh kernel
a1a+a1a.t	1 (+test set)	1605 (+30956)	2	tanh kernel
PolyDistH57	0.1	3000	2	Hausdorff distance
Catcortex	0.8	65	4	cortical connexion strength
Protein	0.8	213	4	sequence-alignment similarity
CoilYork	0.8	288	4	Graph matching
Chicken15-45	0.8	446	5	weighted edit distance between images contours
Chicken29-45	0.8	446	5	weighted edit distance between images contours
Zongker	0.25	2000	10	template matching on handwritten digits
Prodom	0.25	2604	4	pairwise structural alignment on proteins
Chromo-ABS	0.1	4200	22	edit distance on chromosomes
AuralSonar	0.8	100	2	similarity based on human perception
Amazon47	0.8	204	47	similarity based on sells
Patrol	0.8	241	8	similarity based on human memory
Mirex07	0.8	3090	10	similarity based on human evaluation

TABLE 3
This tables gives an overview of the different datasets used in our study. For each, we give the proportion of the dataset that was used for training (p), the dataset size (n), the number of classes (k), and the origin of the indefiniteness. The introduction of the indefiniteness can be quite artificial (like the use of the tanh kernel or toy dissimilarity datasets), induced by the structure of the objects (graphs, strings) or naturally arising in the definition of the task (like similarities based on human perceptions).

6 Discussion and Conclusion

As already mentioned, we are convinced that apart from optimization issues, there is no data-driven reason to enforce positive-definiteness in kernel methods. Literature is replete with examples of applications using indefinite similarities. Even though some attention have been given to this problem in the last years, most of the proposed approaches are deeply linked to the underlying idea that indefiniteness should be corrected, or at least hidden in some space-embeddings. The effects of such methods are:

1) the addition of training parameters (the "amount" of correction...)
2) the need for the transformation of the test points
3) the potential loss of information
4) to obtain a non optimal solution (in the sense of performance rate)

Dataset	KSVM-L	RVM	LP	Best
Balls3D	**41.37%** (6.67)	47.5% (6.12)	43.62% (5.22)	45.70% [39] (1.7)
diabetes	**22.59%** (2.30)	**22.95%** (3.38)	**22.98 %** (3.47)	**22.92% [14]**
a1a *	**17.24 %** (1.88)	**16.92%** (1.6)	**16.79%** (2.45)	**17.08% [14]** -
a1a+a1a.t	**15.72%**	24.10%	24.05%	**15.6% [41]** ***
PolyDistH57 **	**1.86%** (0.50)	2.92% (0.54)	2.64% (0.61)	5.4% [38] (1.3)
Catcortex	**5.4%** (6.3)	11.53% (8.82)	11.53 (9.8)	7.0% [38] (7.1)
Protein	**0.2 %** (0.7)	**0.2%** (0.7)	2.1% (1.5)	**0.4% [38]** (1.7)
CoilYork	**33.10%** (5.05)	39.05% (6.85)	36.89% (5.63)	**33.6% [39]** (1.2)
Chicken15-45	**6.34%** (2.45)	**7.30%** (2.2)	**7.19%** (2.66)	7% [38] (2.8)
Chicken29-45	**4.6%** (2.5)	7.86% (2.09)	**4.91%** (1.98)	**4.7% [38]** (2.7)
Zongker	5.6% (0.6)	7.8% (0.7)	6.5% (0.6)	**4.4% [38]** (0.6)
Prodom	**0.9%** (0.3)	1.05% (0.4)	1.74 % (0.55)	1.3% [38] (0.5)
Chromo-ABS	**5.3%** (0.3)	5.94% (0.46)	7.7 (0.7)	7.7% [38] (0.4)
AuralSonar	**12,5%** (6.17)	14.5% (8.57)	14.25 (7.99)	**12% [40]** (6)
Amazon47	**12.125%** (7.53)	**12.125 %** (6.85)	**11.87%** (6.22)	15% [4] (4.77)
Patrol	**12.29%** (4.56)	23.33% (9.52)	19.37% (5.81)	**11.56% [4]** (4.54)
Mirex07	**55.59%** (2.23)	58.47 (2.17)	57.59% (2.23)	**55.44% [4]** (2.52)

TABLE 4

Error rates (standard deviations) are obtained on average, based of 20 random splits of each dataset. Results in columns KSVM, RVM and LP are from our experiments, those from column *Best* are extracted from different paper providing experimental results on the same datasets. We took results in [4], [14], [38]–[40]. **the results of IndefiniteSVM may not be on average.* ***when increasing the training set size up to half of the dataset size, the test error goes down to 0.62 (0.16) but we kept the training proportion to 0.1 for the sake of comparison.* ****results with a positive semidefinite kernel*

The following paragraphs summarize the defaults of each of the considered methods used to train indefinite kernel machines.

Why not clip-flip-cut heuristics?

All methods based on the kernel spectrum modification (that has to be done simultaneously for train and test data) suffer from the effects 2,3 and 4 mentioned above. Point 2 is obvious (and trying to ignore the test set transformation leads to disasters as those drawn in section 5.2). Point 3 is linked to the proportion of indefiniteness in the kernel matrix, that can be very large. Point 4 is only linked to point 3 only since solvers are then used in their standard setting.

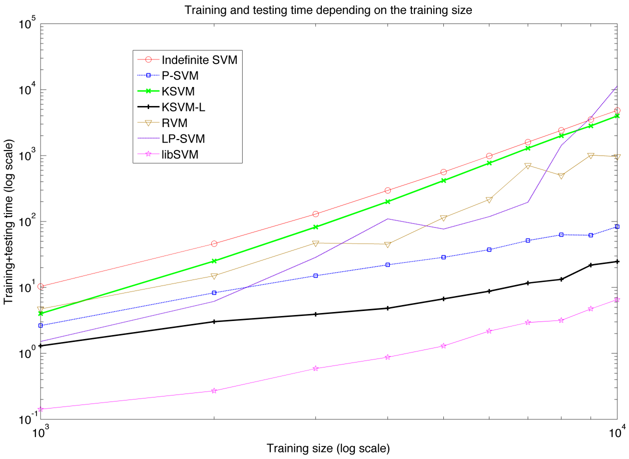

Fig. 5. This figure shows the training and testing time for various methods applicable to indefinite matrices for kernel based classification. The slope of each curve indicates the complexity of the algorithms. The results are obtained for training set size from 1000 to 10000, on checkerboard datasets, which are 2d and separable. The tanh kernel was used with parameters [-1, 1]. The test set's size is 1000.

Why not Indefinite-SVM?

What Indefinite-SVM [14], [18] does is quite similar to the previous case, correcting the kernel in order to suppress or reduce the indefinite part. Doing so it suffers from the same effect (2, 3 and 4) but it also suffers from effect 1, since it requires to cross-validate over a new hyper-parameters linked to the amount of correction to be applied.

Why not non-convex optimization methods?

If one wants to keep the indefinite kernel matrix as it is and still seek for a solution to a SVM-like problem, one clear path is to apply non-convex optimization, like DC-methods [12], [13]. This approach however ignores the Kreĭn space structure. As illustrated in section 5.2, the solutions that are found are not that good in terms of generalization, which is a consequence of the fact that the problem that is actually solved is not the problem that has to be solved. Hence those methods suffer from effect 4. Note here that some authors have pointed out that these methods work well "with sufficiently small C values" [37], which means in other terms, when it is not feasible to optimize towards $-\infty$.

Why not P-SVM?

P-SVM performs least-squares on the kernel entries. It falls in the family of "kernel as features" methods. It can be implemented quite efficiently, but its performances are unstable.

Why not LP-SVM?

LP-SVM have been proposed long ago [42], and very recently in [43], for solving SVM with indefinite matrices. Indeed, applying LP-SVM is actually a way to use the kernel entries as new features: it is a "kernel as features" method. Doing so, the indefiniteness has no more effect, a euclidean distance is actually used. This approach works pretty well in practice and does not suffer from any of the above-mentioned effects. From the series of experiments we have conducted, one could note that the complexity can be an issue: curves on figure 5 are all obtained with Matlab implementations except for LP-SVM, for which we used CPLEX solver since the Matlab one is known to be quite slow. Despite this effort and the well known difference of efficiency between Matlab and compiled languages, LP-SVM exhibits a poor behavior compared to KSVM-L or P-SVM.

Why not RVM?

RVM is based on Bayesian inference, and uses the square of the kernel matrix, which makes it another "kernel as features" method. It has very nice properties such as the absence of the C hyper-parameter or a real sparsity. It also performs quite well most of times on our tests. Note however that we have observed that it tends to perform poorly when the training set size is small.

Why KSVM-L?

Observing the other methods drawbacks does not make the proposed method better by itself. We review here theoretical and practical advantages of KSVM-L:

- KSVM-L solver is the only method so far that uses the specifics of Kreĭn spaces
- the proposed solution lies in the original space
- KSVM-L always performs at least as well as previously proposed ones, in a consistent way: to find equivalent performances for each dataset, we had to pick among more than 10 methods (see results in [4] to observe how difficult it was until now to find a method that works well for a given dataset).
- Among the tested methods, KSVM-L shows the best complexity curve.

Having said that, KSVM-L can still be improved on some practical points:

- The eigen decomposition, even partial, is still troublesome for training time issues and memory issues (it requires to precompute the complete kernel matrix).
- The final solution is not sparse.

6.1 Conclusion

We have shown that solving a stabilization problem instead of a minimization problem for SVM with indefinite matrices has theoretical foundations and leads to better results. We also provide an implementation of our algorithm. We are convinced that KSVM can be successfully applied many application fields, in particular in fields dealing with graph kernels, edit distances and we are currently working on a even more efficient solver.

Acknowledgements

NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

References

[1] C. S. Ong, X. Mary, S. Canu, and A. J. Smola, “Learning with non-positive kernels,” in *ICML '04: Proceedings of the twenty-first international conference on Machine learning*. New York, NY, USA: ACM, 2004, p. 81.

[2] G. Loosli and S. Canu, “Non positive SVM,” in *4th International Workshop on Optimization for Machine Learning,NIPS*, 2011.

[3] J. Laub and K.-R. Müller, “Feature discovery in non-metric pairwise data,” *J. Mach. Learn. Res.*, vol. 5, pp. 801–818, 2004.

[4] Y. Chen, E. K. Garcia, M. R. Gupta, A. Rahimi, and L. Cazzanti, “Similarity-based classification: Concepts and algorithms,” *The Journal of Machine Learning Research*, vol. 10, pp. 747–776, 2009.

[5] F. Suard, A. Rakotomamonjy, and A. Benrshrair, “Kernel on bag of paths for measuring similarity of shapes.” in *ESANN*, 2007, pp. 355–360.

[6] G. Wu, E. Y. Chang, and Z. Zhang, “An analysis of transformation on non-positive semidefinite similarity matrix for kernel machines,” in *Proceedings of the 22nd International Conference on Machine Learning*, vol. 8, 2005.

[7] A. Muñoz and I. M. de Diego, “From indefinite to positive semi-definite matrices,” in *SSPR&SPR 2006*, ser. LNCS 4109, 2006, pp. 764–772.

[8] T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and K. Obermayer, “Classification on pairwise proximity data,” *Advances in neural information processing systems*, pp. 438–444, 1999.

[9] O. L. Mangasarian *et al.*, “Generalized support vector machines,” *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pp. 135–146, 1999.

[10] S. Hochreiter and K. Obermayer, “Support vector machines for dyadic data,” *Neural Computation*, vol. 18, no. 6, pp. 1472–1510, 2006.

[11] M. E. Tipping, “Sparse bayesian learning and the relevance vector machine,” *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Sep. 2001. [Online]. Available: http://dx.doi.org/10.1162/15324430152748236

[12] P.-H. Chen, R.-E. Fan, and C.-J. Lin, “A study on SMO-type decomposition methods for support vector machines,” *Neural Networks, IEEE Transactions on*, vol. 17, no. 4, pp. 893–908, 2006.

[13] P. D. Tao *et al.*, “The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems,” *Annals of Operations Research*, vol. 133, no. 1-4, pp. 23–46, 2005.

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

[15] J. Chen and J. Ye, “Training SVM with indefinite kernels,” in *Proceedings of the 25th International Conference on Machine Learning*. ACM, 2008, pp. 136–143.

[16] Y. Ying, C. Campbell, and M. Girolami, “Analysis of svm with indefinite kernels,” in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, Eds., 2009, pp. 2205–2213.

[17] B. Haasdonk and E. Pekalska, “Indefinite kernel fisher discriminant,” in *International Conference on Pattern Recognition*, 2008. [Online]. Available: http://www.ians.uni-stuttgart.de/publications/2008/HP08a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2477830, IEEE Transactions on Pattern Analysis and Machine Intelligence

[18] Y. Chen, M. R. Gupta, and B. Recht, “Learning kernels from indefinite similarities,” in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 145–152.

[19] S. Gu and Y. Guo, “Learning svm classifiers with indefinite kernels,” in *AAAI*, J. Hoffmann and B. Selman, Eds. AAAI Press, 2012.

[20] H. Sun and Q. Wu, “Least square regression with indefinite kernels and coefficient regularization,” *Applied and Computational Harmonic Analysis*, vol. 30, no. 1, pp. 96 – 109, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1063520310000515

[21] M. Kowalski, M. Szafranski, and L. Ralaivola, “Multiple indefinite kernel learning with mixed norm regularization,” in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML ’09. New York, NY, USA: ACM, 2009, pp. 545–552. [Online]. Available: http://doi.acm.org/10.1145/1553374.1553445

[22] B. Hofmann, P. Mathé, and S. V. Pereverzev, “Regularization by projection: Approximation theoretic aspects and distance functions,” *Journal of Inverse and Ill-posed Problems jiip*, vol. 15, no. 5, pp. 527–545, 2007.

[23] J. Bognár, *Indefinite Inner Product Spaces*. Springer, 1974.

[24] T. Y. Azizov and I. S. Iokhvidov, *Linear Operators in Spaces with an Indefinite Metric*. Wiley, 1989, translated by E. R. Dawson.

[25] B. Hassibi, A. H. Sayed, and T. Kailath, *Indefinite-quadratic estimation and control: a unified approach to H2 and H [infinity] theories*. SIAM, 1999, vol. 16.

[26] T. Ando, “Projections in Kreĭin spaces,” *Linear Algebra and its Applications*, vol. 431, no. 12, pp. 2346 – 2358, 2009, special Issue in honor of Shmuel Friedland. [Online]. Available: http://www.sciencedirect.com/science/article/B6V0R-4W2M6VY-2/2/12e54e1594a427fa9a6f07ece3c93ad3

[27] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[28] R. T. Rockafellar, *Convex Analysis*, reprint ed. Princeton University Press, 1996.

[29] C. Williams and M. Seeger, “The effect of the input density distribution on kernel-based classifiers,” in *International Conference on Machine Learning 17*, P. Langley, Ed. Morgan Kaufmann, 2000, pp. 1159–1166.

[30] F. R. Bach and M. I. Jordan, “Kernel independent component analysis,” *Journal of Machine Learning Research*, vol. 3, pp. 1–48, 2002.

[31] Y. Bengio, O. Delalleau, N. L. Roux, J.-F. Paiement, P. Vincent, and M. Ouimet., “Learning eigenfunctions links spectral embedding and kernel PCA.” *Neural Computation*, vol. 16, no. 10, pp. 2197–2219, 2004.

[32] P. Drineas and M. W. Mahoney, “On the Nyström method for approximating a gram matrix for improved kernel-based learning,” *Journal of Machine Learning Research*, vol. 6, pp. 2153–2175, 2005.

[33] G. Loosli, “SimpleSVM toolbox, 2005.”

[34] T. Knebel, S. Hochreiter, and K. Obermayer, “An SMO algorithm for the potential support vector machine,” *Neural computation*, vol. 20, no. 1, pp. 271–287, 2008.

[35] M. E. Tipping, A. C. Faul *et al.*, “Fast marginal likelihood maximisation for sparse bayesian models,” in *Proceedings of the ninth international workshop on artificial intelligence and statistics*, vol. 1, no. 3, 2003.

[36] ILOG, Inc, “ILOG CPLEX: High-performance software for mathematical programming and optimization,” 2006, see http://www.ilog.com/products/cplex/.

[37] B. Haasdonk, “Feature space interpretation of svms with indefinite kernels,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 482–492, 2005.

[38] E. Pekalska and B. Haasdonk, “Kernel discriminant analysis for positive definite and indefinite kernels,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 6, pp. 1017–1032, 2009.

[39] Duin and Pekalska, “Beyond features: Similarity-based pattern analysis and recognition (SIMBAD),” 2009, deliverable D3.3.

[40] Y. Chen and M. R. Gupta, “Fusing similarities and kernels for classification,” in *Information Fusion, 2009. FUSION’09. 12th International Conference on*. IEEE, 2009, pp. 474–481.

[41] K.-P. Wu and S.-D. Wang, “Choosing the kernel parameters for support vector machines by the inter-cluster distance in the feature space,” *Pattern Recognition*, vol. 42, no. 5, pp. 710–717, 2009.

[42] T. Graepel, R. Herbrich, B. Scholkopf, A. Smola, P. Bartlett, K.-R. Muller, K. Obermayer, and R. Williamson, “Classification on proximity data with LP-machines,” 1999.

[43] I. M. Alabdulmohsin and X. Z. Xin Gao, “Support vector machines with indefinite kernels,” in *Asian Conference on Machine Learning, ACML*, 2014.

Gaëlle Loosli Currently Associate professor at Polytech’Clermont, an engineering school and at Laboratory of Computer Science, Modelisation and Optimization (LIMOS) in Clermont-Ferrand, France. She got her Ph.D. degree from the National institute of applied science in Rouen (INSA) in 2006. Then she joined the National Research Institute of Science and Technology for Environment and Agriculture (IRSTEA) for a postdoctoral position until she joined the LIMOS lab in 2008. She is specialized in kernel methods and machine learning, with applications to 3D shape retrieval.

Stéthane Canu Currently Professor at LITIS research laboratory and at the information technology department, at the National institute of applied science in Rouen (INSA). He received a Ph.D. degree in System Command from Comigne University of Technology in 1986. He joined the faculty department of Computer Science at Compiegne University of Technology in 1987 and the French habilitation degree from Paris 6 University in 1995. In 1997, he joined the Rouen Applied Sciences National Institute (INSA) as a full professor, where he created the information engineering department. His research interests includes kernels machines, regularization, machine learning applied to signal processing, pattern classification, factorization for recommender systems and learning for context aware applications.

Cheng Soon Ong Principal researcher at the Machine Learning Research Group, NICTA. He is also an adjunct associate professor at the Australian National University, and an honourary research fellow at the University of Melbourne. His PhD in Computer Science was completed at the Australian National University in 2005. He was a postdoc at the Max Planck Institute of Biological Cybernetics and the Fredrich Miescher Laboratory in Tübingen, Germany. From 2008 to 2011, he was a lecturer in the Department of Computer Science at ETH Zurich, and he has been with NICTA since 2012. He is interested in enabling scientific discovery by extending statistical machine learning methods. In recent years, he has developed new optimization methods for solving problems such as ranking, feature selection and experimental design, with the aim of solving scientific questions in collaboration with experts in other fields.