Indefinite Kernel Spectral Learning

Siamak Mehrkanoon[1], Xiaolin Huang [2,*], Johan A.K. Suykens[1]

[1] *Department of Electrical Engineering (ESAT-STADIUS), KU Leuven, Kasteelpark Arenberg 10, B-3001, Leuven, Belgium*

[2] *Institute of Image Processing and Pattern Recognition, and the MOE Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, 200240, Shanghai, P.R. China*

Abstract

The use of indefinite kernels has attracted many research interests in recent years due to their flexibility. They do not possess the usual restrictions of being positive definite as in the traditional study of kernel methods. This paper introduces the indefinite unsupervised and semi-supervised learning in the framework of least squares support vector machines (LS-SVM). The analysis is provided for both unsupervised and semi-supervised models, i.e., Kernel Spectral Clustering (KSC) and Multi-Class Semi-Supervised Kernel Spectral Clustering (MSS-KSC). In indefinite KSC models one solves an eigenvalue problem whereas indefinite MSS-KSC finds the solution by solving a linear system of equations. For the proposed indefinite models, we give the feature space interpretation, which is theoretically important, especially for the scalability using Nyström approximation. Experimental results on several real-life datasets are given to illustrate the efficiency of the proposed

Email addresses: siamak.mehrkanoon@esat.kuleuven.be (Siamak Mehrkanoon[1]), xiaolinhuang@sjtu.edu.cn (Xiaolin Huang [2,*]), johan.suykens@esat.kuleuven.be (Johan A.K. Suykens[1])

indefinite kernel spectral learning.

Keywords: Semi-supervised learning, scalable models, indefinite kernels, kernel spectral clustering, low embedding dimension

1. Introduction

Kernel-based learning models have shown great success in various application domains [28, 29, 30]. Traditionally, kernel learning is restricted to positive semi-definite (PSD) kernels as the properties of Reproducing Kernel Hilbert Spaces (RKHS) are well explored. However, many positive semi-definite kernels such as the sigmoid kernel [1] remain positive semi-definite only when their associated parameters are within a certain range, otherwise they become non-positive definite [2]. Moreover, the positive definite kernels are limited in some problems due to the need of non-Euclidean distances [3, 27]. For instance in protein similarity analysis, the protein sequence similarity measures require learning with a non-PSD similarity matrix [4].

The need of using indefinite kernels in machine learning methods attracted many research interests on indefinite learning in both theory and algorithm. Theoretical discussions are mainly on Reproducing Kernel Kreĭn Spaces (RKKS, [5, 6]), which is different to the RKHS for PSD kernels. In algorithm design, a lot of attempts have been made to cope with indefinite kernels by regularizing the non-positive definite kernels to make them positive semi-definite [7, 8, 9, 10]. It is also possible to directly use an indefinite kernel in e.g., support vector machine (SVM) [1]. Though an indefinite kernel makes the problem non-convex, it is still possible to get a local optimum as suggested by [11]. One important issue is that kernel trick is no longer

valid when an indefinite kernel is applied in SVM and one needs new feature space interpretations to explain the effectiveness of SVM with indefinite kernels. The interpretation is usually about a pseudo-Euclidean (pE) space, which is a product of two Euclidean vector spaces, as analyzed in [12] and [6]. Notice that "indefinite kernels" literally covers asymmetric ones and complex ones. But this paper restricts "indefinite kernel" to the kernels that correspond to real symmetric indefinite matrices, which is consistent to the existing literature on indefinite kernel.

Indefinite kernels are also applicable to the least squares support vector machines [13]. In LS-SVM, one solves a linear system of equations in the dual and the optimization problem itself has no additional requirement on the positiveness of the kernel. In other words, even if an indefinite kernel is used in the dual formulation of LS-SVM, it is still convex and easy to solve, which is different from indefinite kernel learning with SVM. However, like in SVM, using an indefinite kernel in LS-SVM looses the traditional interpretation of the feature space and a new formulation has been recently discussed in [14].

Motivated by the success of indefinite learning for some supervised learning tasks, we in this paper introduce indefinite similarities to unsupervised as well as semi-supervised models that can learn from both labeled and unlabeled data instances. There have been already many efficient semi-supervised models, such as Laplacian support vector machine [15], which assumes that neighboring point pairs with a large weight edge are most likely within the same cluster. However, to the best of our knowledge, there is no work that extends unsupervised/semi-supervised learning to indefinite kernels.

Since using indefinite kernels in the framework of LS-SVM does not change the training problem, here we focus on multi-class semi-supervised kernel spectral clustering (MSS-KSC) model proposed by Mehrkanoon et al. [16]. MSS-KSC model and its extensions for analyzing large-scale data, data streams as well as multi-label datasets are discussed in [17, 18, 19] respectively. When one of the regularization parameters is set to zero, MSS-KSC becomes the kernel spectral clustering (KSC), which is an unsupervised learning algorithm introduced by [20]. It is a special case of MSS-KSC. Due to the link to LS-SVM, it can be expected and also will be shown here that MSS-KSC with indefinite similarities are still easy to solve. However, the kernel trick is no longer valid and we have to find corresponding feature space interpretations. The purpose of this paper is to introduce indefinite kernels for semi-supervised learning as well as unsupervised learning as a special case. Specifically, we propose indefinite kernels in MSS-KSC and KSC models. Subsequently, we derive their feature space interpretation. Besides of theoretical interests, the interpretation allows us to develop algorithms based on Nyström approximation for large-scale problems.

The paper is organized as follows. Section 2 briefly reviews the MSS-KSC with PSD kernel. In Section 3, the MSS-KSC with an indefinite kernel is derived and the interpretation of the feature map is provided. As a special case of MSS-KSC, the KSC with an indefinite kernel and its feature interpretation is discussed in Section 4. In Section 5, we discuss the scalability of the indefinite KSC/MSS-KSC model on large-scale problems. The experimental results are given in Section 6 to confirm the validity and applicability of the proposed model on several real life small and large-scale datasets. Section 7

ends the paper with a brief conclusion.

2. MSS-KSC with PSD kernel

Consider training data

$$\mathcal{D} = \{\underbrace{x_1, ..., x_{n_{UL}}}_{\substack{\text{Unlabeled} \\ (\mathcal{D}_U)}}, \underbrace{x_{n_{UL}+1}, .., x_n}_{\substack{\text{Labeled} \\ (\mathcal{D}_L)}}\}, \tag{1}$$

where $\{x_i\}_{i=1}^n \in \mathbb{R}^d$. The first n_{UL} points do not have labels whereas the last $n_L = n - n_{UL}$ points have been labeled. Assume that there are Q classes ($Q \leq N_c$), then the label indicator matrix $Y \in \mathbb{R}^{n_L \times Q}$ is defined as follows:

$$Y_{ij} = \begin{cases} +1 & \text{if the } i\text{th point belongs to the } j\text{th class,} \\ -1 & \text{otherwise.} \end{cases} \tag{2}$$

The primal formulation of multi-class semi-supervised KSC (MSS-KSC) described by [16] is given as follows:

$$\begin{aligned} \min_{w^{(\ell)}, b^{(\ell)}, e^{(\ell)}} \quad & \frac{1}{2}\sum_{\ell=1}^{Q} w^{(\ell)^T} w^{(\ell)} - \frac{\gamma_1}{2}\sum_{\ell=1}^{Q} e^{(\ell)^T} V e^{(\ell)} + \\ & \frac{\gamma_2}{2}\sum_{\ell=1}^{Q} (e^{(\ell)} - c^{(\ell)})^T \tilde{A} (e^{(\ell)} - c^{(\ell)}) \\ \text{subject to} \quad & e^{(\ell)} = \Phi w^{(\ell)} + b^{(\ell)} 1_n, \ \ell = 1, \ldots, Q, \end{aligned} \tag{3}$$

where c^ℓ is the ℓ-th column of the matrix C defined as

$$C = [c^{(1)}, \ldots, c^{(Q)}]_{n \times Q} = \left[\frac{0_{n_{UL} \times Q}}{Y}\right]_{n \times Q}. \tag{4}$$

Here

$$\Phi = [\varphi(x_1), \ldots, \varphi(x_n)]^T \in \mathbb{R}^{n \times h}$$

where $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^h$ is the feature map and h is the dimension of the feature space which can be infinite dimensional. $0_{n_{UL} \times Q}$ is a zero matrix of size $n_{UL} \times Q$, Y is defined previously, and the right hand of (4) is a matrix consisting of $0_{n_{UL} \times Q}$ and Y. The matrix $\tilde{A}$ is defined as follows:

$$\tilde{A} = \left[\begin{array}{c|c} 0_{n_{UL} \times n_{UL}} & 0_{n_{UL} \times n_L} \\ \hline 0_{n_L \times n_{UL}} & I_{n_L \times n_L} \end{array} \right],$$

where $I_{n_L \times n_L}$ is the identity matrix of size $n_L \times n_L$. V is the inverse of the degree matrix defined as follows:

$$V = D^{-1} = \text{diag}\left(\frac{1}{d_1}, \cdots, \frac{1}{d_n}\right),$$

where $d_i = \sum_{j=1}^{n} K(x_i, x_j)$ is the degree of the i-th data point.

As stated in [16], the object function in the formulation (3), contains three terms. The first two terms together with the set of constraints correspond to a weighted kernel PCA formulation in the least squares support vector machine framework given in [20] which is shown to be suitable for clustering and is referred to as kernel spectral clustering (KSC) algorithm. The last regularization term in (3) aims at minimizing the squared distance between the projections of the labeled data and their corresponding labels. This term enforces the projections of the labeled data points to be as close as possible to the true labels. Therefore by incorporating the labeled information, the pure clustering KSC model is guided so that it respects the provided labels by not misclassifying them. In this way, one could learn from both labeled and unlabeled instances. In addition thanks to introduced model selection scheme in [16], the MSS-KSC model is also equipped with the out-of-sample extension property to predict the labels of unseen instances.

It should be noted that, ignoring the last regularization term, or equivalently setting $\gamma_2 = 0$ and $Q = N_c - 1$, reduces the MSS-KSC formulation to kernel spectral clustering (KSC) described in [20]. Therefore, KSC formulation in the primal can be covered as a special case of MSS-KSC formulation. As illustrated by [16], given Q labels the approach is not restricted to finding just Q classes and instead is able to discover up to 2^Q hidden clusters. In addition, it uses a low embedding dimension to reveal the existing number of clusters which is important when one deals with large number of clusters.

When the feature map φ in (3) is not explicitly known, in the context of PSD kernel, one may use the kernel trick and solve the problem in the dual. Elimination of the primal variables $w^{(\ell)}, e^{(\ell)}$ and making use of Mercer's Theorem result in the following linear system in the dual [16]:

$$\gamma_2 \left(I_n - \frac{R1_n 1_n^T}{1_n^T R 1_n} \right) c^{(\ell)} = \alpha^{(\ell)} - R \left(I_n - \frac{1_n 1_n^T R}{1_n^T R 1_n} \right) \Omega \alpha^{(\ell)}, \tag{5}$$

where $R = \gamma_1 V - \gamma_2 \tilde{A}$. In (5), there are two coefficients, namely γ_1 and γ_2, which reflect the emphasis on unlabeled and labeled samples, respectively, as shown in (3). Besides, there could be one or multiple parameters in the kernel. All of these parameters could be tuned by cross-validation.

3. MSS-KSC with Indefinite Kernel

Traditionally, the kernel used in MSS-KSC is restricted to be positive semi-definite. When the kernel in (5) is indefinite, one still requires to solve a linear system of equations. However, the feature space has different interpretations compared to definite kernels. In what follows we establish and analyze the feature space interpretations for MSS-KSC.

Theorem 3.1. *Suppose that for a symmetric but indefinite kernel matrix K, the solution of the linear system (5) is denoted by $[\alpha_*, b_*]^T$. Then there exist two feature mappings φ_1 and φ_2, which correspond to the matrices Φ_1 and Φ_2, respectively, such that*

$$w_1^{(\ell)} = \sum_{i=1}^{n} \alpha_{*,i}^{(\ell)} \, \varphi_1(x_i), \ell = 1, \ldots, Q, \tag{6}$$

and

$$w_2^{(\ell)} = \sum_{i=1}^{n} \alpha_{*,i}^{(\ell)} \, \varphi_2(x_i), \ell = 1, \ldots, Q, \tag{7}$$

which is a stationary point of the following primal problem:

$$\begin{aligned} \min_{w_1^{(\ell)}, w_2^{(\ell)}, b^{(\ell)}, e^{(\ell)}} \quad & \frac{1}{2} \sum_{\ell=1}^{Q} w_1^{(\ell)^T} w_1^{(\ell)} - \frac{1}{2} \sum_{\ell=1}^{Q} w_2^{(\ell)^T} w_2^{(\ell)} \\ & + \frac{\gamma_2}{2} \sum_{\ell=1}^{Q} (e^{(\ell)} - c^{(\ell)})^T \tilde{A} (e^{(\ell)} - c^{(\ell)}) - \frac{\gamma_1}{2} \sum_{\ell=1}^{Q} e^{(\ell)^T} V e^{(\ell)} \\ \text{subject to} \quad & e^{(\ell)} = \Phi_1 w_1^{(\ell)} + \Phi_2 w_2^{(\ell)} + b_*^{(\ell)} 1_n, \ell = 1, \ldots, Q. \end{aligned} \tag{8}$$

Then, the dual problem of (8) is given in (5), with the kernel matrix Ω defined as follows,

$$\Omega_{i,j} = K_1(x_i, x_j) - K_2(x_i, x_j), \tag{9}$$

where, $K_1(x_i, x_j)$ and $K_2(x_i, x_j)$ are two PSD kernels.

Proof. The Lagrangian of the constrained optimization problem (8) becomes

$$\begin{aligned} \mathcal{L}(w_1^{(\ell)}, w_2^{(\ell)}, b_*^{(\ell)}, e^{(\ell)}, \alpha_*^{(\ell)}) = & \frac{1}{2} \sum_{\ell=1}^{Q} w_1^{(\ell)^T} w_1^{(\ell)} - \sum_{\ell=1}^{Q} w_2^{(\ell)^T} w_2^{(\ell)} - \frac{\gamma_1}{2} \sum_{\ell=1}^{Q} e^{(\ell)^T} V e^{(\ell)} \\ + \frac{\gamma_2}{2} \sum_{\ell=1}^{Q} (e^{(\ell)} - c^{(\ell)})^T \tilde{A} (e^{(\ell)} - c^{(\ell)}) & + \sum_{\ell=1}^{Q} \alpha_*^{(\ell)^T} \left(e^{(\ell)} - \Phi_1 w_1^{(\ell)} - \Phi_2 w_2^{(\ell)} - b_*^{(\ell)} 1_n \right), \end{aligned}$$

where $\alpha_*^{(\ell)}$ is the vector of Lagrange multipliers. Then the KKT optimality conditions are as follows,

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w_1^{(\ell)}} = 0 \rightarrow \ w_1^{(\ell)} = \Phi_1^T \alpha_*^{(\ell)}, \ \ell = 1, \ldots, Q, \\ \frac{\partial \mathcal{L}}{\partial w_2^{(\ell)}} = 0 \rightarrow \ w_2^{(\ell)} = -\Phi_2^T \alpha_*^{(\ell)}, \ \ell = 1, \ldots, Q, \\ \frac{\partial \mathcal{L}}{\partial b^{(\ell)}} = 0 \rightarrow \ 1_n^T \alpha_*^{(\ell)} = 0, \ \ell = 1, \ldots, Q, \\ \frac{\partial \mathcal{L}}{\partial e^{(\ell)}} = 0 \rightarrow \ \alpha_*^{(\ell)} = (\gamma_1 V - \gamma_2 \tilde{A}) e^{(\ell)} + \gamma_2 c^{(\ell)}, \ \ell = 1, \ldots, Q, \\ \frac{\partial \mathcal{L}}{\partial \alpha_*^{(\ell)}} = 0 \rightarrow \ e^{(\ell)} = \Phi_1 w_1^{(\ell)} + \Phi_2 w_2^{(\ell)} + b_*^{(\ell)} 1_n, \ \ell = 1, \ldots, Q. \end{cases} \tag{10}$$

Elimination of the primal variables $w_1^{(\ell)}, w_2^{(\ell)}, e^{(\ell)}$ and making use of the kernel trick ($\Omega_1 = \Phi_1^T \Phi_1$ and $\Omega_2 = \Phi_2^T \Phi_2$) lead to the linear system of equations in the dual defined in (5) with the indefinite kernel matrix defined in (9). With α_* obtained from (5), the weight vectors $w_1^{(\ell)}$ and $w_2^{(\ell)}$ defined in (6) and (7), satisfy the first-order optimality condition of (8).

□

One can show that from the third KKT optimality condition, the bias term is determined by

$$b_*^{(\ell)} = (1/1_n^T R 1_n)(-1_n^T \gamma_2 c^{(\ell)} - 1_n^T R \Omega \alpha_*^{(\ell)}), \ell = 1, \ldots, Q, \tag{11}$$

where R is defined as in (5). Once the solution vector and the bias term are obtained, one can use the out-of-sample extension property of the model to predict the score variables of the unseen test instances as follows:

$$e_{\text{test}}^{(\ell)} = \Omega \alpha_*^{(\ell)} + b_*^{(\ell)}, \ell = 1, \ldots, Q. \tag{12}$$

The above discussion gives the feature space interpretation for indefinite MSS-KSC. The discussion in a pE space is similar to indefinite SVM; see, [6]

[12] and [14]. The main difference from learning algorithms for PSD kernels is that the indefinite learning is to minimize a pseudo-distance. The readers are referred to Fig. 1 in [12], which givs a clear geometric explanation for the distance in a pE space.

In practice, the performance of the MSS-KSC model depends on the choice of the parameters. In this aspect, there is no difference between a PSD kernel and an indefinite kernel. Therefore the following model selection scheme introduced in [16] for MSS-KSC can be employed:

$$\max_{\gamma_1,\gamma_2,\mu} \quad \eta\,\mathrm{Sil}(\gamma_1,\gamma_2,\mu) + (1-\eta)\mathrm{Acc}(\gamma_1,\gamma_2,\mu). \tag{13}$$

It is a combination of Silhouette (Sil) and classification accuracy (Acc). $\eta \in [0,1]$ is a user-defined parameter that controls the trade off between the importance given to unlabeled and labeled instances. The MSS-KSC algorithm with an indefinite kernel is summarized in Algorithm 1. One can note that the main difference with respect to Algorithm 1 discussed in [16] is at the level employing the indefinite kernel and all the other steps remain unchanged.

4. KSC with indefinite kernels - as a special case

As a special case of MSS-KSC formulation (8), when $\gamma_2 = 0$ and $Q = N_c - 1$, we obtain (17), i.e., the KSC model given by [20]. This dual problem itself does not require the positiveness of Ω. Thus, an indefinite kernel is applicable here and one still solves an eigenvalue problem. However, the kernel trick, which is the key to build primal-dual relationship for definite kernels, cannot be used for indefinite kernels, which follows that different

Algorithm 1 Indefinite kernel in multi-class semi-supervised classification model

1: **Input:** Training data set $\mathcal{D}$, labels Z, tuning parameters $\{\gamma_i\}_{i=1}^{2}$, kernel parameter μ, test set $\mathcal{D}^{\text{test}} = \{x_i^{\text{test}}\}_{i=1}^{N_{\text{test}}}$ and codebook $\mathcal{CB} = \{c_q\}_{q=1}^{Q}$

2: **Output:** Class membership of test data $\mathcal{D}^{test}$

3: Construct the indefinite kernel matrix Ω (see (9)).

4: Solve the dual linear system (5) with the indefinite kernel matrix Ω to obtain $\{\alpha^{\ell}\}_{\ell=1}^{Q}$ and compute the bias term $\{b_*^{\ell}\}_{\ell=1}^{Q}$ using (11).

5: Estimate the test data projections $\{e_{\text{test}}^{(\ell)}\}_{\ell=1}^{Q}$ using (12).

6: Binarize the test projections and form the encoding matrix $[\text{sign}(e_{test}^{(1)}), \ldots, \text{sign}(e_{\text{test}}^{(Q)})]_{N_{\text{test}} \times Q}$ for the test points (Here $e_{\text{test}}^{(\ell)} = [e_{test,1}^{(\ell)}, \ldots, e_{test,N_{\text{test}}}^{(\ell)}]^T$).

7: For each i, assign x_i^{test} to class q^*, where $q^* = \underset{q}{\text{argmin}}\, d_H(e_{\text{test},i}^{(\ell)}, c_q)$ and $d_H(\cdot,\cdot)$ is the Hamming distance.

feature space interpretations are needed. In this section, we establish and analyze the feature space interpretations, similar to the discussion for indefinite MSS-KSC.

Theorem 4.1. *Suppose that the solution of the eigenvalue problem (17), in the dual, for a symmetric but indefinite kernel matrix K is denoted by $[\alpha_*, b_*]^T$. Then there exist two feature mappings φ_1 and φ_2, such that*

$$w_1^{(\ell)} = \sum_{i=1}^{n} \alpha_{*,i}^{(\ell)}\, \varphi_1(x_i), \ell = 1, \ldots, N_c - 1, \tag{14}$$

and

$$w_2^{(\ell)} = \sum_{i=1}^{n} \alpha_{*,i}^{(\ell)}\, \varphi_2(x_i), \ell = 1, \ldots, N_c - 1, \tag{15}$$

which is a stationary point of the following primal problem:

$$\begin{aligned} \min_{w_1^{(\ell)}, w_2^{(\ell)}, b_*^{(\ell)}, e^{(\ell)}} \quad & \frac{1}{2}\sum_{\ell=1}^{N_c-1} w_1^{(\ell)^T} w_1^{(\ell)} - \frac{1}{2}\sum_{\ell=1}^{N_c-1} w_2^{(\ell)^T} w_2^{(\ell)} - \frac{\gamma_1}{2}\sum_{\ell=1}^{N_c-1} e^{(\ell)^T} V e^{(\ell)} \\ \text{subject to} \quad & e^{(\ell)} = \Phi_1 w_1^{(\ell)} + \Phi_2 w_2^{(\ell)} + b_*^{(\ell)} 1_n,\ \ell = 1, \ldots, N_c - 1. \end{aligned} \tag{16}$$

Then, the dual problem of (16) is given as:

$$V P_v \Omega \alpha^{(\ell)} = \lambda \alpha^{(\ell)}, \tag{17}$$

where $\lambda = n/\gamma_\ell$, $\alpha^{(\ell)}$ *are the Lagrange multipliers and* P_v *is the weighted centering matrix:*

$$P_v = I_n - \frac{1}{1_n^T V 1_n} 1_n 1_n^T V.$$

Here I_n *is the* $n \times n$ *identity matrix and the kernel matrix* Ω *is defined as follows,*

$$\Omega_{i,j} = K_1(x_i, x_j) - K_2(x_i, x_j), \tag{18}$$

where, $K_1(x_i, x_j)$ *and* $K_2(x_i, x_j)$ *are two PSD kernels.*

Proof. It follows the proof of indefinite MSS-KSC model described in (3).

□

From the link between KSC and LS-SVM, the above theorem also could be regarded as a weighted and multi-class extension of the result obtained by [14]. To give an intuitive idea that using indefinite kernels in KSC is possible, we show a simple example that applies the truncated ℓ_1 distance (TL1) kernel [21], which is indefinite and takes the following formulation,

$$K(s, t) = \max\{\mu - \|s - t\|_1, 0\}. \tag{19}$$

For this problem, one can observe that KSC with an indefinite kernel indeed can successfully cluster the points, as shown in Figure 1. Here the Silhouette index is used for model selection (see [22] for overview of the internal clustering quality metrics).

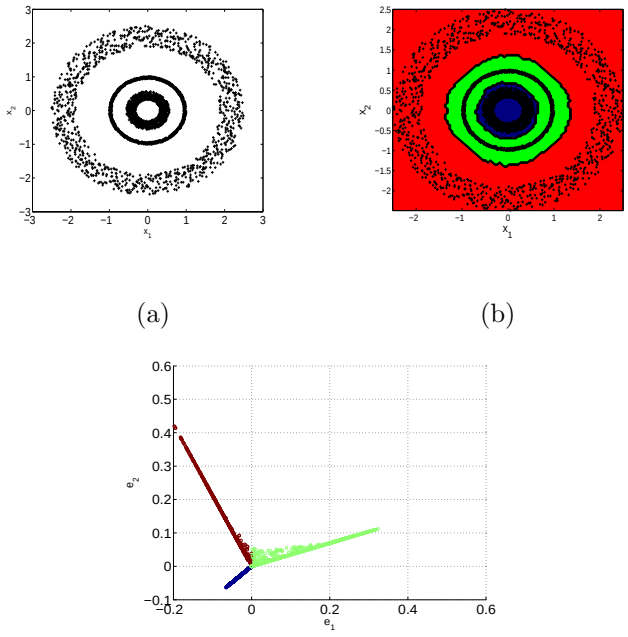


Figure 1: Illustrating the performance of KSC model with an indefinite kernel (TL1 kernel) on synthetic three concentric example. (a) Original data. (b) The predicted memberships obtained using indefinite KSC model with $\mu = 0.4$. (c) The line structure of the score variables, e, indicating the good generalization performance of indefinite KSC model with $\mu = 0.4$.

Theorem 4.1 and Theorem 4.2 are both based on the positive decomposition of an indefinite kernel matrix Ω: since it is a symmetric and real matrix,

we can surely find two PSD matrices K_1 and K_2 such that

$$\Omega_{ij} = K_{1ij} - K_{2ij}.$$

For example, K_1 and K_2 can be constructed from the positive and negative eigenvalues of Ω. This decomposition indicates that a PSD kernel is a special case of indefinite kernel with $K_{2ij} = 0$. Therefore, the use of indefinite kernel in spectral learning provides flexibility to improve the performance of PSD learning, if the kernel, which could be indefinite or definite, is suitably designed.

5. Scalability

Kernel based models have shown to be successful in many machine learning tasks. However, unfortunately, many of them scale poorly with the training data size due to the need for storing and computing the kernel matrix which is usually dense.

In the context of kernel based semi-supervised learning with PSD kernels, attempts have been made to make the kernel based models scalable, see [17, 23, 24]. Mehrkanoon, et al. [17] introduced the Fixed-Size MSS-KSC (FS-MSS-KSC) model for classification of large-scale partially labeled instances. FS-MSS-KSC uses an explicit feature map approximated by the Nyström method [13, 25] and solves the optimization problem in the primal. The finite dimensional approximation of the feature map is obtained by numerically solving a Fredholm integral equation using the Nyström discretization method which results in an eigenvalue decomposition of the kernel matrix Ω; see [25].

The i-th component of the n-dimensional feature map $\hat{\varphi} : \mathbb{R}^d \to \mathbb{R}^n$, for any point $x \in \mathbb{R}^d$, can be obtained as follows:

$$\hat{\varphi}_i(x) = \frac{1}{\sqrt{\lambda_i^{(s)}}} \sum_{k=1}^{n} u_{ki}\, K(x_k, x), \tag{20}$$

where $\lambda_i^{(s)}$ and u_i are eigenvalues and eigenvectors of the kernel matrix $\Omega_{n\times n}$. Furthermore, the k-th element of the i-th eigenvector is denoted by u_{ki}. In practice when n is large, we work with a subsample (prototype vectors) of size $m \ll n$ of which the elements are selected using an entropy based criterion. In this case, the m-dimensional feature map $\hat{\varphi} : \mathbb{R}^d \to \mathbb{R}^m$ can be approximated as follows:

$$\hat{\varphi}(x) = [\hat{\varphi}_1(x), \ldots, \hat{\varphi}_m(x)]^T, \tag{21}$$

where

$$\hat{\varphi}_i(x) = \frac{1}{\sqrt{\lambda_i^{(s)}}} \sum_{k=1}^{m} u_{ki}\, K(x_k, x), i = 1, \ldots, m. \tag{22}$$

Here, $\lambda_i^{(s)}$ and u_i are the eigenvalues and eigenvectors of the constructed kernel matrix $\Omega_{m\times m}$ with the selected prototype vectors.

When an indefinite kernel is used, the matrix K has both positive and negative eigenvalues. Thus, according to the previous feature interpretations, one can then construct two approximations for the feature maps Φ_1 and Φ_2 based on positive and negative eigenvalues, respectively. Here we give the following lemma to explain the approximation for indefinite MSS-KSC and a similar result is valid for indefinite KSC as well.

Lemma 5.1. *Given the m-dimensional approximation to the feature map, i.e. $\hat{\Phi}_1 = [\hat{\varphi}(x_1), \ldots, \hat{\varphi}(x_n)]^T \in \mathbb{R}^{n\times m_1}$, and $\hat{\Phi}_2 = [\hat{\varphi}(x_1), \ldots, \hat{\varphi}(x_n)]^T \in$*

$\mathbb{R}^{n\times m_2}$, *and regularization constants* $\gamma_1, \gamma_2, \in \mathbb{R}^+$, *the solution to (8) is obtained by solving the following linear system of equations in the primal:*

$$\left[\begin{array}{c|c|c} \hat{\Phi}_1^T R \hat{\Phi}_1 + I_{m_1} & \hat{\Phi}_1^T R \hat{\Phi}_2 & \hat{\Phi}_1^T R 1_n \\ \hline \hat{\Phi}_2^T R \hat{\Phi}_1 & \hat{\Phi}_1^T R \hat{\Phi}_1 - I_{m_2} & \hat{\Phi}_2^T R 1_n \\ \hline 1_n{}^T R \hat{\Phi}_1 & 1_n{}^T R \hat{\Phi}_2 & 1_n{}^T R 1_n \end{array}\right] \left[\begin{array}{c} w_1^{(\ell)} \\ \hline w_2^{(\ell)} \\ \hline b^{(\ell)} \end{array}\right] = \gamma_2 \left[\begin{array}{c} \hat{\Phi}_1^T c^{(\ell)} \\ \hline \hat{\Phi}_2^T c^{(\ell)} \\ \hline 1_n^T c^{(\ell)} \end{array}\right], \ \ell = 1, \ldots, Q, \tag{23}$$

where $R = \gamma_2 A - \gamma_1 V$ *is a diagonal matrix,* V *and* R *are given previously.* I_{m_1} *and* I_{m_2} *are the identity matrix of size* $m_1 \times m_1$ *and* $m_2 \times m_2$ *respectively.*

Proof. Substituting the explicit feature maps $\hat{\Phi}_1$ and $\hat{\Phi}_2$ into formulation (8), one can rewrite it as an unconstrained optimization problem. Subsequently setting the derivative of the cost function with respect to the primal variables $w_1^{(\ell)}$, $w_2^{(\ell)}$ and $b^{(\ell)}$ to zero results in the linear system (23). □

The score variables evaluated at the test set $\mathcal{D}^{\text{test}} = \{x_i\}_{i=1}^{n_{\text{test}}}$ become:

$$e_{\text{test}}^{(\ell)} = \hat{\Phi}_1^{\text{test}} w_1^{(\ell)} + \hat{\Phi}_2^{\text{test}} w_2^{(\ell)} + b^{(\ell)} 1_{n_{\text{test}}} \ \ell = 1, \ldots, Q, \tag{24}$$

where $\hat{\Phi}_1^{\text{test}} = [\hat{\varphi}(x_1), \ldots, \hat{\varphi}(x_{n_{\text{test}}})]^T \in \mathbb{R}^{n_{\text{test}}\times m_1}$ and $\hat{\Phi}_2^{\text{test}} = [\hat{\varphi}(x_1), \ldots, \hat{\varphi}(x_{n_{\text{test}}})]^T \in \mathbb{R}^{n_{\text{test}}\times m_2}$. The decoding scheme consists of comparing the binarized score variables for test data with the codebook $\mathcal{CB}$ and selecting the nearest codeword in terms of Hamming distance.

6. Numerical Experiments

In this section, experimental results on a synthetic as well as several real-life datasets from the UCI machine learning repository [26] are given. We also show the applicability of the proposed indefinite method on a simple

image segmentation task. Furthermore, the performance of the model for classification of partially labeled large-scale datasets using indefinite kernels will be studied in this section.

The performance of kernel learning relies on the choice of kernel. In this paper, we consider two indefinite kernels in KSC/MSS-KSC. One is the TL1 kernel (19) and the other is the tanh kernel with parameters c, d:

$$K(s,t) = \tanh(cs^T t + d). \tag{25}$$

Notice that when $c > 0$, the tanh kernel is conditionally positive definite; otherwise, it is indefinite. In the following experiments, c is selected from both positive and negative vales, and hence the tanh kernel is regarded as an indefinite kernel in this paper. The performance of these indefinite kernels will be compared with the RBF kernel, which is the most popular PSD kernel and takes the following formulation:

$$K(s,t) = \exp(-\|s-t\|_2^2/\sigma^2). \tag{26}$$

6.1. Semi-supervised classification

First, Two-moons dataset, a 2-dimensional synthetic problem, is considered to visualize the performance of indefinite kernels in a semi-supervise setting. The results obtained via the RBF kernel and the TL1 kernel are shown in Fig.2, from which it can be seen that the two classes have been successfully classified by both the PSD and the non-PSD kernel. One may notice that the decision boundaries obtained by the TL1 kernel is not as smooth as the RBF kernel. It is due to the piecewise linearity of the TL1 kernel and could be different if other non-PSD kernels are used.

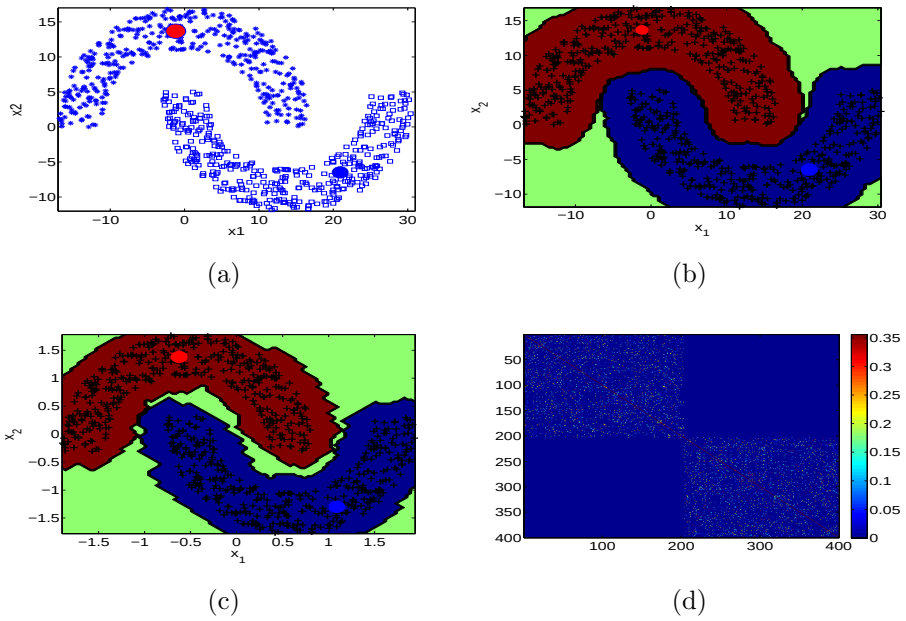


Figure 2: Illustrating the performance of MSS-KSC model on synthetic single labeled example. (a) Original labeled and unlabeled points. (b) The predicted memberships obtained using MSS-KSC model with the RBF kernel. (c) The predicted memberships obtained using MSS-KSC model with an indefinite kernel. (d) The associated similarity matrix indicating the cluster structure in the data.

Next, we conduct experiments on real-life datasets from UCI repository [26]. Here, 60% of the whole data (at random) is used as test set and the remaining 40% as training set. We randomly select part of the training data as the labeled and the remaining ones as the unlabeled training data. The ratio of the labeled training data points that is used in our experiments is denoted as follows:

$$\text{ratio}_{\text{label}} = \frac{\#\text{ labeled training data points}}{\#\text{ training data points}}.$$

The considered ratios for forming a labeled training set are one-fourth, one-third and half of the whole training dataset. To reduce the randomness of the experiment, we repeat this process 10 times. At each run, 10-fold cross validation is performed for model selection. The parameters to tune are the regularization constants γ_1, γ_2 and kernel parameters. In our experiments, we set $\gamma_1 = 1$ and then find reasonable values for γ_2, μ in the range $[10^{-3}, 10^0]$ and $[0, d]$, respectively. For the RBF kernel, and $\sigma \in \{10^{-4}, 10^{-3}, \ldots, 10^4\}$. For tanh kernel, the candidate sets are $c \in \{-0.5 - 0.2, -0.1, 0, 0.1, 0.2, 0.5\}$ and $d \in \{2^{-10}, 2^{-7}, \ldots, 2^3\}$. The cross-validation performance on the Wine dataset for the TL1 kernel is shown in Fig.3, from which and other experiments, we empirically observed that the TL1 kernel enjoys good stability on its kernel parameters. This makes its performance for a pre-given value, e.g., $\mu = 0.7d$, satisfactory in many tested examples.

The average accuracy on the test dataset over 10 trials are reported in Table 1, where the details of the datasets are provided as well. From the results, one can observe that the performance of the unsupervised KSC model with an indefinite kernel is generally comparable to that with the RBF kernel. For most problems, the TL1 kernel with a pre-given μ outputs good results.

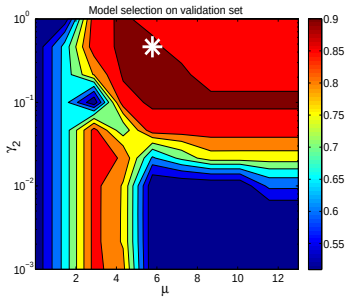


Figure 3: Illustrating the sensitivity of the MSS-KSC model with respect to its parameters, γ_2 and μ in the case of the TL1 kernel for the Wine dataset.

Moreover, there are indeed some problems, like Monk3 and Ionosphere, for which indefinite kernel learning can improve the performance significantly.

6.2. Clustering

The experimental results on several real world clustering datasets[1] using KSC model with the RBF and the TL1 kernel are reported in Table 2. The cluster memberships of these datasets are not known beforehand, therefore the clustering results can be evaluated by internal clustering quality metrics such as the widely used silhouette index (Sil-index) and the Davies Bouldin index (DB-index) [22]. Larger values of Sil-index imply better clustering quality. While, the lower the value of DB-index means that the clustering quality is better. In Table 2, the best indices are underlined where one can observe the good performance of the TL1 kernel. Notice that simply from these experiments, we cannot conclude indefinite kernel is better or worse than the definite ones. But the results indicate that for some problems, it

[1]http://cs.joensuu.fi/sipu/datasets/ (Accessed: 2015-12-29)

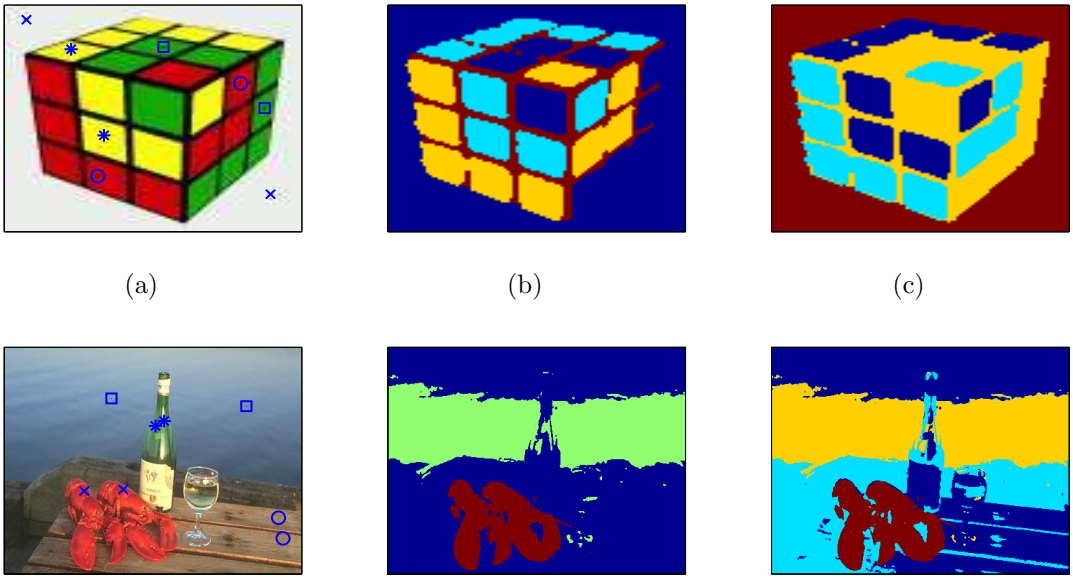
is worth to consider the proposed indefinite unsupervised learning methods, which may further improve the performance from the traditional PSD kernel learning methods.

6.3. Image segmentation

Here we show the application of the proposed indefinite Kernel on unsupervised and semi-supervised image segmentation. Following the lines of [18], for each image, a local color histogram with a 5×5 local window around each pixel is computed using minimum variance color quantization of eight levels. A subset of 500 unlabeled pixels together with some labeled pixels are used for training and the whole image for test. The original and labeled images together with segmentation results are shown in Fig 4. One can qualitatively observe that thanks to the provided labeled pixels, the semi-supervised model performs better than completely unsupervised model on the test images.

6.4. Large scale datasets

Here we show the possibility of applying the TL1 kernel in the context of semi-supervised learning on large-scale datasets. The size of the real-life data, on which the experiments were conducted, ranges from medium to large and covering both binary and multi-class classification. The classification of these datasets is performed using different number of training labeled and unlabeled data instances. In our experiments, for all the datasets, 20% of the whole data (at random) is used for test, and the training set is constructed from the reaming 80% of the data. In order to have a realistic setting, the number of unlabeled training points are considered to be p times more than that of labeled training points, where, in our experiments, depending on the



(a) (b) (c)

(d) (e) (f)

Figure 4: Illustrating the performance of MSS-KSC model with an indefinite kernel (TL1) on image segmentation. (a,d) The labeled images. (b,e) The segmentations obtained by unsupervised KSC model with the TL1 kernel. (c,f) The segmentation obtained by semi-supervised MSS-KSC model with the TL1 kernel.

size of the dataset, p ranges from 2 to 5. Descriptions of the considered datasets can be found in Table 3.

The average results of the proposed MSS-KSC model with the TL1 kernel together with that of Fixed-size MSS-KSC [17] are tabulated in Table 4. From Table 4, one can observe that the proposed MSS-KSC algorithm with an indefinite kernel has been successfully applied on large scab data and its accuracy is comparable to that of the RBF kernel. This is an interesting point as in many applications one need to address the scalability of the models when using indefinite kernel. It should be mentioned that as expected, the computational time of MSS-KSC with the RBF kernel is faster than that of MSS-KSC with the TL1 kernel. This can be explained by the fact that in the RBF kernel, one feature map is constructed where as in the TL1 kernel one needs to calculate two feature maps.

7. Conclusions

Motivated by success of indefinite kernels in supervised learning, we in this paper proposed to use indefinite kernels in the semi-supervised learning framework. Specifically, we studies the indefinite KSC and MSS-KSC models. For both models the optimization problems remain easy to solve if indefinite kernels are used. The interpretations of the feature map in the case of indefinite kernels are provided. Based on these interpretations, Nyström approximation can be used for the scalability of indefinite KSC and MSS-KSC. The proposed indefinite learning methods are evaluated on real datasets in comparison with the existing methods with the RBF kernel. One can observe that for some datasets, the indefinite kernel shows its superiority,

which implies that there are some semi-supervised tasks requiring indefinite learning methods. For example, when some (dis)similarity induces to indefinite kernels, it is better to directly use those indefinite kernel rather than to find approximate PSD ones. Furthermore, if an indefinite kernel is suitably selected or designed, the indefinite learning performance could be very promising.

Acknowledgments

The authors are grateful to the anonymous reviewer for insightful comments.

The research leading to these results received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC AdG A-DATADRIVE-B (290923). This letter reflects only our views: The EU is not responsible for any use that may be made of the information in it. The research leading to these results received funds from the following sources: Research Council KUL: GOA/10/09 MaNet, CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants; Flemish Government: FWO: PhD /Postdoc grants, projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); IWT: PhD/Postdoc grants, projects: SBO POM (100031); iMinds Medical Information Technologies SBO 2014; Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017). Siamak Mehrkanoon was supported by a Postdoctoral Fellowship of the Research Foundation-Flanders (FWO). Xiaolin Huang is supported by National Natural Science Foundation of China (no. 61603248). Johan Suykens is a full professor at KU Leuven, Belgium.

References

[1] V. Vapnik, Statistical Learning Theory, 1998, Wiley.

[2] Q. Wu, Regularization networks with indefinite kernels, Journal of Approximation Theory 166 (2013) 1–18.

[3] E. Pekalska, B. Haasdonk, Kernel discriminant analysis for positive definite and indefinite kernels, IEEE Trans. on Pattern Analysis and Machine Intelligence, 31 (2009) 1017–1032.

[4] Y. Chen, M. R. Gupta, B. Recht, Learning kernels from indefinite similarities, in: proceedings of the 26th International Conference on Machine Learning, 2009, pp. 145–152.

[5] C. S. Ong, X. Mary, S. Canu, A. J. Smola, Learning with non-positive kernels, in: proceedings of the 21st International Conference on Machine Learning, 2004, pp. 639–646.

[6] G. Loosli, S. Canu, C. S. Ong, Learning SVM in Kreĭn spaces, IEEE Trans. on Pattern Analysis and Machine Intelligence, 38 (2016) 1204–1216.

[7] E. Pekalska, P. Paclik, R. P. W. Duin, A generalized kernel approach to dissimilarity-based classification. Journal of Machine Learning Research 2 (2002) 175–211.

[8] R. Luss, A. d'Aspremont, Support vector machine classification with indefinite kernels, in: Advances in Neural Information Processing Systems, 2008, pp. 953–960.

[9] J. Chen, J. Ye, Training SVM with indefinite kernels, in: proceedings of the 25th International Conference on Machine Learning, 2008, pp. 136–143.

[10] Y. Ying, C. Campbell, M. Girolami, Analysis of SVM with indefinite kernels, in: Advances in Neural Information Processing Systems, 2009, pp. 2205–2213.

[11] H.-T. Lin, C.-J. Lin, A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods, 2003, internal report.

[12] B. Haasdonk, Feature space interpretation of SVMs with indefinite kernels, IEEE Transactions on Pattern Analysis and Machine Intelligence, 27 (2005) 482–492.

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

[14] X. Huang, A. Maier, J. Hornegger, J. A. K. Suykens, Indefinite kernels in least squares support vector machine and kernel principal component analysis, Applied and Computational Harmonic Analysis, 43 (2017), 162–172.

[15] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, The Journal of Machine Learning Research 7 (2006) 2399–2434.

[16] S. Mehrkanoon, C. Alzate, R. Mall, R. Langone, J. A. K. Suykens, Multiclass semisupervised learning based upon kernel spectral clustering, IEEE Trans. Neural Networks and Learning Systems 26 (2015), 720–733.

[17] S. Mehrkanoon, J. A. K. Suykens, Large scale semi-supervised learning using KSC based model, in: International Joint Conference on Neural Networks, 2014, pp. 4152–4159.

[18] S. Mehrkanoon, O. M. Agudelo, J. A. K. Suykens, Incremental multi-class semi-supervised clustering regularized by Kalman filtering, Neural Networks 71 (2015) 88–104.

[19] S. Mehrkanoon, J. A. K. Suykens, Multi-label semi-supervised learning using regularized kernel spectral clustering, in: International Joint Conference on Neural Networks, 2016, pp. 4009–4016.

[20] C. Alzate, J. A. K. Suykens, Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA, IEEE Trans. on Pattern Analysis and Machine Intelligence 32 (2010) 335–347.

[21] X. Huang, J. A. K. Suykens, S. Wang, A. Maier, J. Hornegger, Classification with truncated ℓ_1 distance kernel, IEEE Transactions on Neural Networks and Learning Systems doi:10.1109/TNNLS.2017.2668610..

[22] J. C. Bezdek, N. R. Pal, Some new indexes of cluster validity, IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics, 28 (1998) 301–315.

[23] G. S. Mann, A. McCallum, Simple, robust, scalable semi-supervised learning via expectation regularization, in: proceedings of the 24th International Conference on Machine Learning, 2007, pp. 593–600.

[24] W. Liu, J. He, S.-F. Chang, Large graph construction for scalable semi-supervised learning, in: proceedings of the 27th International Conference on Machine Learning, 2010, pp. 679–686.

[25] C. Williams, M. Seeger, Using the Nyström method to speed up kernel machines, in: Advances in Neural Information Processing Systems, 2001, pp. 682–688.

[26] A. Asuncion, D. J. Newman, UCI Machine Learning Repository (2007).

[27] F.M. Schleif, and P. Tino, Indefinite Core Vector Machine, Pattern Recognition Volume 71, November 2017, pp. 187–195.

[28] D. Wang, and X. Zhang, and M. Fan, and X. Ye, Hierarchical mixing linear support vector machines for nonlinear classification, Pattern Recognition Volume 59, 2016, pp. 255–267.

[29] Y. Li, and X. Tian, and M. Song, and D. Tao, Multi-task proximal support vector machine, Pattern Recognition Volume 48, 2015, pp. 3249–3257.

[30] J. Richarz, and S. Vajda, and R. Grzeszick, and G.A. Fink, Semi-supervised learning for character recognition in historical archive documents, Pattern Recognition Volume 47, 2014, pp. 1011–1020.

Table 1: The average accuracy and the standard deviation of the LapSVMp [15] and MSS-KSC on the test set using PSD and indefinite kernels.

					MSS-KSC Method				
Dataset	d	Q	$\text{ratio}_{\text{label}}$	$\mathcal{D}^{\text{train}}_{\text{Labeled}}/\mathcal{D}^{\text{train}}_{\text{Unlabeled}}/\mathcal{D}^{\text{test}}$	**RBF Kernel** σ **is tuned**	**TL1 Kernel** μ **is tuned**	**TL1 Kernel** $\mu = 0.7d$	**Tanh-Kernel** c, d **is tuned**	**LapSVMp**
Iris	4	3	1/4	15/45/90	0.85 ± 0.09	0.88 ± 0.07	0.86 ± 0.09	0.65 ± 0.11	0.70 ± 0.12
			1/3	20/40/90	0.87 ± 0.07	0.88 ± 0.09	0.86 ± 0.03	0.71 ± 0.07	0.76 ± 0.11
			1/2	30/30/90	0.92 ± 0.03	0.90 ± 0.08	0.88 ± 0.09	0.77 ± 0.10	0.83 ± 0.10
Wine	13	3	1/4	18/54/106	0.89 ± 0.07	0.90 ± 0.08	0.89 ± 0.03	0.59 ± 0.12	0.73 ± 0.11
			1/3	24/48/106	0.92 ± 0.01	0.93 ± 0.01	0.92 ± 0.03	0.75 ± 0.11	0.84 ± 0.09
			1/2	36/36/106	0.94 ± 0.01	0.95 ± 0.02	0.93 ± 0.03	0.84 ± 0.12	0.90 ± 0.10
Zoo	16	7	1/4	11/30/60	0.89 ± 0.05	0.84 ± 0.10	0.75 ± 0.17	0.60 ± 0.10	0.78 ± 0.08
			1/3	14/27/60	0.89 ± 0.04	0.90 ± 0.04	0.80 ± 0.10	0.66 ± 0.09	0.82 ± 0.11
			1/2	21/20/60	0.90 ± 0.04	0.89 ± 0.04	0.83 ± 0.17	0.72 ± 0.12	0.85 ± 0.10
Seeds	7	3	1/4	21/63/126	0.87 ± 0.05	0.88 ± 0.03	0.85 ± 0.09	0.62 ± 0.10	0.80 ± 0.10
			1/3	28/56/126	0.88 ± 0.09	0.86 ± 0.09	0.85 ± 0.04	0.70 ± 0.12	0.83 ± 0.11
			1/2	42/42/126	0.90 ± 0.01	0.88 ± 0.02	0.88 ± 0.02	0.79 ± 0.11	0.87 ± 0.09
Monk1	6	2	1/4	56/167/333	0.63 ± 0.04	0.66 ± 0.03	0.63 ± 0.03	0.59 ± 0.09	0.60 ± 0.10
			1/3	75/148/333	0.67 ± 0.03	0.69 ± 0.03	0.64 ± 0.03	0.60 ± 0.03	0.65 ± 0.11
			1/2	112/111/333	0.68 ± 0.07	0.70 ± 0.08	0.70 ± 0.03	0.63 ± 0.07	0.69 ± 0.08
Monk2	6	2	1/4	61/180/360	0.63 ± 0.08	0.61 ± 0.06	0.54 ± 0.03	0.57 ± 0.02	0.58 ± 0.11
			1/3	81/160/360	0.64 ± 0.06	0.62 ± 0.05	0.55 ± 0.03	0.61 ± 0.06	0.63 ± 0.10
			1/2	121/120/360	0.71 ± 0.04	0.65 ± 0.06	0.58 ± 0.02	0.63 ± 0.03	0.66 ± 0.11
Monk3	6	2	1/4	56/166/332	0.74 ± 0.03	0.81 ± 0.03	0.81 ± 0.02	0.68 ± 0.10	0.77 ± 0.08
			1/3	74/148/332	0.79 ± 0.02	0.85 ± 0.03	0.83 ± 0.04	0.74 ± 0.02	0.80 ± 0.09
			1/2	111/111/332	0.81 ± 0.02	0.87 ± 0.03	0.87 ± 0.02	0.77 ± 0.04	0.84 ± 0.10
Pima	8	2	1/4	77/231/460	0.70 ± 0.01	0.70 ± 0.03	0.70 ± 0.03	0.62 ± 0.14	0.70 ± 0.08
			1/3	74/148/460	0.71 ± 0.02	0.72 ± 0.03	0.71 ± 0.01	0.69 ± 0.02	0.71 ± 0.10
			1/2	154/154/460	0.72 ± 0.02	0.72 ± 0.02	0.72 ± 0.02	0.70 ± 0.05	0.72 ± 0.06
Ionosphere	33	2	1/4	36/105/210	0.77 ± 0.05	0.81 ± 0.08	0.75 ± 0.07	0.69 ± 0.04	0.77 ± 0.09
			1/3	47/94/210	0.83 ± 0.06	0.88 ± 0.03	0.77 ± 0.07	0.71 ± 0.05	0.83 ± 0.08
			1/2	71/70/210	0.86 ± 0.07	0.88 ± 0.03	0.79 ± 0.05	0.73 ± 0.03	0.86 ± 0.09

Table 2: Comparison of the KSC model with PSD and indefinite kernel, K-means and landmark-based spectral clustering algorithm using two internal clustering quality metrics, i.e. Silhouette and DB index, on some real datasets.

				Silhouette index			**DB index**		
Dataset	n	d	N_c	**RBF**	**TL1**	**K-means**	**RBF**	**TL1**	**K-means**
Wine	178	13	3	0.44	0.46	<u>0.50</u>	1.41	<u>1.06</u>	1.22
Thyroid	215	3	2	0.68	<u>0.81</u>	0.75	0.52	<u>0.43</u>	0.97
Breast	699	9	2	0.75	0.75	0.75	<u>0.77</u>	0.86	0.76
Glass	214	9	7	0.81	<u>0.84</u>	0.63	1.20	<u>1.09</u>	0.64
Iris	150	4	3	0.77	0.77	0.64	0.73	<u>0.59</u>	0.70

Table 3: Dataset statistics

Dataset	# points	# attributes	# classes
Adult	48,842	14	2
IJCNN	141,691	22	3
Cod-RNA	331,152	8	2
Covertype	581,012	54	3
SUSY	5,000,000	18	2
Sensorless	58,509	48	11
letter	20,000	16	26
Satimage	6,435	36	6
texture	5,500	40	11
USPS	9,298	256	10

Table 4: Comparing the average test accuracy, standard deviation and computation time of the FS-MSS-KSC model [17] with the RBF kernel and the TL1 kernel on real-life datasets over 10 simulation runs.

						Test Accuracy		**Computation time (in seconds)**	
Dataset	p	$\text{ratio}_{\text{label}}$	$\mathcal{D}_{\text{tr}}^{L}$	$\mathcal{D}_{\text{tr}}^{U}$	$\mathcal{D}_{\text{test}}$	**RBF**	**TL1**	**RBF**	**TL1**
USPS	2	1/3	1,000	2,000	1859	$\underline{0.86 \pm 0.002}$	$\underline{0.86 \pm 0.002}$	$\underline{0.02}$	0.16
		1/3	2,000	4,000	1859	0.88 ± 0.003	$\underline{0.89 \pm 0.002}$	$\underline{0.02}$	0.81
texture	3	1/4	500	1,500	1,100	0.85 ± 0.002	$\underline{0.87 \pm 0.002}$	$\underline{0.01}$	0.02
		1/4	1,000	3,000	1,100	0.89 ± 0.004	$\underline{0.91 \pm 0.001}$	$\underline{0.02}$	0.05
Satimage	3	1/4	500	1,500	1,287	0.83 ± 0.003	$\underline{0.85 \pm 0.003}$	$\underline{0.01}$	0.02
		1/4	1,000	3,000	1,287	0.85 ± 0.001	$\underline{0.86 \pm 0.002}$	$\underline{0.02}$	0.05
Adult	3	1/4	4,000	12,000	9,768	0.844 ± 0.003	$\underline{0.847 \pm 0.006}$	$\underline{0.08}$	0.20
		1/4	8,000	24,000	9,768	0.846 ± 0.003	$\underline{0.852 \pm 0.005}$	$\underline{0.22}$	0.34
letter	3	1/4	2,000	6,000	4,000	0.65 ± 0.002	$\underline{0.68 \pm 0.003}$	$\underline{0.05}$	0.12
		1/4	4,000	12,000	4,000	0.69 ± 0.004	$\underline{0.71 \pm 0.002}$	$\underline{0.12}$	0.25
Sensorless	3	1/4	4,000	12,000	11,701	0.92 ± 0.002	$\underline{0.93 \pm 0.002}$	$\underline{0.24}$	1.46
		1/4	8,000	24,000	11,701	0.94 ± 0.001	$\underline{0.96 \pm 0.001}$	$\underline{0.54}$	3.21
IJCNN	5	1/6	4,000	20,000	28,338	$\underline{0.935 \pm 0.004}$	0.933 ± 0.001	$\underline{0.51}$	1.70
		1/6	16,000	80,000	28,338	$\underline{0.956 \pm 0.002}$	0.953 ± 0.001	$\underline{2.53}$	6.01
Cod-RNA	5	1/6	8,000	40,000	66,230	$\underline{0.959 \pm 0.001}$	0.952 ± 0.001	$\underline{0.92}$	1.57
		1/6	32,000	160,000	66,230	$\underline{0.962 \pm 0.0005}$	0.958 ± 0.001	$\underline{6.63}$	8.27
Covertype	5	1/6	8,000	40,000	116,202	0.732 ± 0.001	$\underline{0.740 \pm 0.003}$	$\underline{1.80}$	8.01
		1/6	64,000	320,000	116,202	$\underline{0.781 \pm 0.001}$	0.772 ± 0.002	$\underline{12.20}$	25.7
SUSY	2	1/3	500,000	1,000,000	1,000,000	$\underline{0.771 \pm 0.001}$	$\underline{0.771 \pm 0.001}$	$\underline{4.91}$	15.98
		1/3	1,000,000	2,000,000	1,000,000	0.783 ± 0.001	$\underline{0.787 \pm 0.001}$	$\underline{10.01}$	34.70